



UNIVERSITAT DE
BARCELONA

Trabajo Final de Grado

GRADO DE INGENIERÍA INFORMÁTICA

**Facultad de Matemáticas e Informática
Universidad de Barcelona**

Gestión de incidencias multiplataforma

Álvaro Escurra González

Director: Josep Vañó Chic
Realizado en: Departamento de
Matemáticas e Informática
Barcelona, 27 de junio de 2018

AGRADECIMIENTOS

Ante todo, me gustaría agradecer especialmente a mi familia por todo el apoyo que he recibido desde el primer hasta el último día en estos últimos años. El esfuerzo y aliento de superación que me han ofrecido ha merecido la pena. Sin ellos, este objetivo personal no lo hubiera conseguido. También, agradecer a mi tutor del TFG y a todos los profesores de la carrera por todo lo que he aprendido de cada uno de ellos.

Por último, a mis compañeros de la carrera y mis compañeros de trabajo que han hecho que me inspire en este proyecto.

Gracias, este trabajo va por ustedes. ¡Lo conseguimos!

RESUMEN

Este proyecto Final de Grado consiste en el desarrollo de una aplicación web para gestionar incidencias en empresas que ofrecen Servicio de Atención Técnica (SAT). En este documento se describirá todo el análisis, el diseño y la implementación que se ha llevado a cabo. Además, cuenta con la puesta en marcha de la aplicación en un entorno real.

Constantemente, las tecnologías van cambiando y es por esto por lo que en este proyecto se ha apostado por el desarrollo de una aplicación web, donde en la implementación se hace uso del *framework Laravel 5.6 de PHP* y, para el desarrollo de las páginas, se ha utilizado *HTML5, CSS3 y JavaScript*, además de todas las tecnologías que la envuelven, como por ejemplo *AJAX* que nos permite tener una aplicación dinámica.

La aplicación permite que los clientes, en este caso usuarios, puedan entrar en la aplicación y registrar nuevas incidencias, así como consultar el proceso que se lleva a cabo al actuar sobre una de ellas. También, en el lado de la administración, la aplicación permite la gestión de las incidencias que llevarán a cabo los técnicos, teniendo un claro impacto en la productividad y optimización del trabajo.

Las funcionalidades que proporciona la aplicación se centran en: la gestión de incidencias, permitiendo crear, modificar, asignar y resolver las incidencias rápidamente; también, permite gestionar el calendario de los técnicos a través de registrar, en un calendario compartido, aspectos como: reuniones, desplazamientos, vacaciones, etc. Por otra parte, también se centra en la gestión de clientes, la cual permite registrar nuevos clientes o modificar su información básica y su localización; finalmente, la gestión de usuarios y perfiles para que puedan acceder a la aplicación. Cabe destacar que la aplicación ofrece notificaciones, tanto por correo electrónico como dentro de la misma aplicación, para que el cliente sepa en cada momento el transcurso de su incidencia, así como las comunicaciones que puede tener con el técnico.

Por último, respecto al diseño, este se caracteriza por ser atractivo ya que permite que la usabilidad del usuario sea cómoda y fácil. Además, se trata de un diseño *responsive* que permite su uso desde cualquier dispositivo móvil.

RESUM

Aquest projecte Final de Grau consisteix en el desenvolupament d'una aplicació web per a gestionar incidències a empreses que ofereixen Servei d'Atenció Tècnica (*SAT*). En aquest document es descriurà tot l'anàlisi, el disseny i la implementació que s'ha portat a terme. A més, compta amb la posada en marxa de l'aplicació a un entorn real.

Constantment, les tecnologies pateixen canvis i és per aquest motiu que aquest projecte aposta pel desenvolupament d'una aplicació web, on per a la seva implementació es fa ús del *framework Laravel 5.6 de PHP* y, pel desenvolupament de les pàgines, s'ha utilitzat *HTML, CSS3 i JavaScript*, a més de totes les tecnologies que l'envolten, com per exemple *AJAX* que ens permet tenir una aplicació dinàmica.

L'aplicació permet que els clients, en aquest cas els usuaris, puguin accedir dins l'aplicació i enregistrar noves incidències, així com consultar el procés que es porta a terme en el moment d'actuar sobre aquestes. També, per part de l'administració, l'aplicació permet la gestió de les incidències que portaran a terme els tècnics, fet que tindrà un clar impacte en la productivitat i l'optimització del treball.

Les funcionalitats que proporciona l'aplicació es centren en: la gestió d'incidències, permetent crear, modificar, assignar i resoldre les incidències ràpidament; també, permet gestionar el calendari dels tècnics a través del registre, a un calendari compartit, d'aspectes com: reunions, desplaçaments, vacances, etc. D'altra banda, també es centra en la gestió de clients, el qual permet enregistrar nous clients o modificar la seva informació bàsica i la seva localització. Cal destacar que l'aplicació ofereix notifikacions, tant per correu electrònic com dins la mateixa aplicació, amb la finalitat que el client sàpiga en cada moment el transcurs de la seva incidència, així com les comunicacions que pot tenir amb el tècnic.

Finalment, respecte al disseny, aquest es caracteritza pel seu atractiu ja que permet que la usabilitat de l'usuari sigui còmode i fàcil. A més, es tracta d'un disseny *responsive* que permet el seu ús des de qualsevol dispositiu mòbil.

ABSTRACT

This Final Degree Project consists of the development of a web application to manage incidents in companies that offer Technical Attention Service (SAT). This document will describe all the analysis, design and implementation that has been carried out. In addition, it has the implementation of the application in a real environment.

Technology is in constant change, hence why this project has opted for the development of a web application, where the backend implementation has been made using the *Laravel 5.6 PHP* framework and, for the frontend development, *HTML5*, *CSS3* and *JavaScript* have been chosen, as well as all the technologies that surround it, such as *AJAX*, which allows us to have a dynamic application.

The application allows, on one side, for the customers to report incidents and review the state and actions being carried on them and, on the other side, for the technicians to coordinate the response for the reported incidents, thus having a positive impact on productivity and work optimization.

The functionalities provided by the application are focused on: incident management, allowing to create, modify, assign and solve incidents quickly; also, it allows to manage the calendar of the technicians through registering, in a shared calendar, aspects such as: meetings, displacements, vacations, etc. On the other hand, it also focuses on customer management, which allows registering new clients or modifying their basic information and location; finally, the management of users and profiles so that they can access the application.

It should be noted that the application offers notifications, both by email and within the same application, so that the customer knows at all times the course of its occurrence, as well as communications that may have with the technician.

Finally, regarding the visual design, its focus on ease of use and comfortability allows for an attractive experience for the end user. In addition, its responsive design allows the application to being used on any mobile device.

Índice

1. INTRODUCCIÓN.....	8
1.1 MOTIVACIÓN Y CONTEXTO DEL PROYECTO.....	8
1.2 DESCRIPCIÓN DEL PROYECTO	10
1.3 OBJETIVOS.....	10
1.3.1 <i>Objetivos generales</i>	10
1.3.2 <i>Objetivos específicos</i>	10
1.4 METODOLOGÍA.....	11
1.5 PLANIFICACIÓN TEMPORAL	12
1.5.1 <i>Distribución del tiempo</i>	12
1.5.2 <i>Diagrama de Gantt</i>	13
2. ANÁLISIS DEL SISTEMA.....	14
2.1 REQUISITOS INICIALES	14
2.1.1 <i>Requisitos funcionales</i>	14
2.1.2 <i>Requisitos no funcionales</i>	16
2.2 CASOS DE USO	16
2.2.1 <i>Diagrama de casos de uso</i>	17
2.2.2 Descripción textual de casos de uso	19
3. DISEÑO	32
3.1 DECISIONES TECNOLÓGICAS.....	32
3.2 ARQUITECTURA DEL SOFTWARE.....	32
3.2.1 Modelo	33
3.2.2 Vista.....	34
3.2.3 Controlador.....	35
3.3 DIAGRAMA DE CLASES	37
3.4 PROTOTIPOS.....	38
4. IMPLEMENTACIÓN	40
4.1 EL MODELO	40
4.2 SISTEMA DE AUTENTICACIÓN	41
Autenticación.....	42
4.3 MIDDLEWARE	42
4.4 USABILIDAD Y DISEÑO ADAPTATIVO	43
4.5 SISTEMA DE CRUD	46
4.6 SUBIR FICHEROS	48
4.7 LLAMADAS ASÍNCRONAS AJAX	49
4.8 SEGURIDAD	50
4.9 PERMISOS.....	51

4.10 EVENTOS / NOTIFICACIONES.....	51
<i>Notificaciones</i>	52
4.11 COMPONENTES DE LA APLICACIÓN	53
Listados.....	53
Buscadores.....	54
Widgets	54
Calendario.....	55
Pop-ups	55
5. PRUEBAS Y RESULTADOS.....	56
6. CONCLUSIONES.....	59
6.1 FUTURAS AMPLIACIONES	59
7. GLOSARIO	61
8. BIBLIOGRAFÍA.....	62
ANEXO. MANUAL DE USUARIO	63
<i>Manual para usuarios SAT</i>	63
<i>Manual para clientes</i>	71

1. Introducción

1.1 Motivación y contexto del proyecto

Este proyecto surge como una mejora de la planificación y organización de las tareas diarias que se realizan en el día a día en el contexto de una empresa dedicada al servicio de mantenimiento informático.

Las empresas que ofrecen este servicio pueden llegar a tener múltiples incidencias al día; las cuales tienen diferentes niveles de prioridad (desde mínima hasta urgentes), o incluso incidencias que requieren desplazamiento de los técnicos hasta la localización del cliente. Si a esto le añadimos que se trata de una empresa con una gran cartera de clientes, el volumen de incidencias se multiplica. Por lo tanto, se convierte en una necesidad tener un buen sistema capaz de gestionarlas y organizarlas, asignando a los empleados encargados de resolverlas; permitiendo, de esta manera, obtener la mayor productividad de los trabajadores y, en consecuencia, asegurar la mayor satisfacción de los clientes.

Generalmente, existen muchas aplicaciones de software dedicadas a cubrir estas necesidades, pero se trata de sistemas complejos con un sinnúmero de características que complican su utilidad. Además, la mayoría limitan su acceso desde diferentes dispositivos, es decir, son softwares de escritorio, con lo cual no permiten su uso fuera de la oficina.

La forma de trabajar está cambiando y se necesitan aplicaciones que puedan responder de forma ágil a los cambios y necesidades de las empresas, y que faciliten la gestión de sus demandas. Debido a esto, este proyecto consiste en optimizar la respuesta de este tipo de software para que éste sea una herramienta sencilla capaz de conectarse desde diferentes dispositivos, facilitando la comunicación y el desarrollo de las tareas.

Después de investigar sobre los diferentes softwares existentes con esta finalidad, he encontrado que presentan diferentes problemas por los cuales surge la necesidad de optimizarlos. Uno de los problemas más comunes que se encuentran en las empresas que disponen de este sistema es que el software queda obsoleto por el paso del tiempo y de las nuevas tecnologías en potencia.

Adentrándome en un caso concreto, una empresa dedicada al servicio de mantenimiento informático que he podido analizar ha tenido este problema y, en consecuencia, este problema ha obligado a que sus trabajadores tengan una comunicación vía correo electrónico sobre las

actuaciones que llevan a cabo sobre una incidencia, hecho que a largo plazo ralentiza o dificulta el trabajo. Para que esto se pueda entender mejor, voy a especificar el ejemplo de esta empresa:

Esta empresa, recibe un número considerable de llamadas al día que son atendidas por los mismos técnicos. Se trata de llamadas de clientes que, o bien quieren comentar nuevas incidencias que se deben llevar a cabo, o bien son llamadas preguntado sobre el estado de su incidencia (que previamente los técnicos han recogido vía mail). El problema se encuentra en que, si las incidencias son recogidas vía mail, estos correos de forma frecuente quedan trasapelados en la bandeja de entrada del correo. Este es un problema bastante grave, ya que debemos recurrir a la búsqueda del correo específico, cosa que puede originar una gran pérdida de tiempo y los resultados de optimización y productividad no son siempre los deseados.

Otro problema derivado de la gestión de las incidencias vía correo electrónico se puede dar a la hora de reasignar incidencias. Es decir, se puede dar el caso de que el técnico principal asignado a la incidencia no haya podido finalizarla y, por X motivo, se vea obligado a traspasarla a otro técnico. En este caso, al no tener ningún registro sobre ella, la comunicación sobre las actuaciones que se han llevado a cabo es o bien por vía mail o bien de forma oral, corriendo el riesgo de perder detalles o información relevante que, en consecuencia, harán que el nuevo técnico asignado tenga que llamar al cliente nuevamente para que le resuelva las dudas al respecto o, incluso, para que le indique dónde continuar.

Por último, existe la posibilidad de que una incidencia se convierta, también, en un evento programado para una fecha en concreto. Habitualmente, para cubrir esta necesidad, la empresa hace uso del calendario de Outlook. Una de las limitaciones que este recurso presenta es que no es posible consultar los eventos programados desde otros dispositivos. En consecuencia, si un técnico tiene programada una visita, muchas veces los demás compañeros no tienen constancia de ello, por lo tanto, desconocen su ubicación y su hora de llegada; teniendo un impacto negativo en la organización del equipo de trabajo.

Como se ha comentado antes, la empresa hace uso de una aplicación que ha quedado obsoleta. En ella se pueden picar las incidencias, pero resulta incómodo la manera de introducirlas, poco intuitiva y no llega a satisfacer los problemas que surgen actualmente en el día a día de los trabajadores.

Personalmente, la realización de este proyecto supone un reto, dada la complejidad que implica optimizar la organización del trabajo y el manejo de los diferentes usuarios en una misma plataforma destinada a una empresa con una cartera de clientes elevada.

1.2 Descripción del proyecto

El proyecto consiste en una aplicación web multiplataforma que, gracias a una mejora de la organización, optimizará la gestión de incidencias llevada a cabo por el equipo de técnicos dentro de un entorno empresarial.

Este programa permitirá la introducción de nuevas incidencias, ya sea desde el espacio proporcionada para sus clientes como el de los mismos técnicos, si fuera necesario, permitiendo el registro de todos los estados por los que pasa una incidencia hasta el momento de su resolución. De manera simultánea, también establece comunicación con los clientes que serán notificados sobre el estado de su incidencia.

Para acceder a la aplicación se requiere de autenticación (usuario y contraseña). Esta permitirá el acceso a un espacio donde el usuario tendrá disponible diferentes contenidos, que dependerán de su perfil asignado.

Por otra parte, podrán trabajar desde cualquier dispositivo móvil o escritorio, desde donde tendrán acceso a todas las funcionalidades de la aplicación: podrán consultar sus tareas, los eventos programados, realizar gestiones sobre las incidencias, acceder a las fichas del cliente, buscar y ver los históricos de las incidencias, etc.

1.3 Objetivos

1.3.1 Objetivos generales

El objetivo principal de la aplicación es mejorar la productividad del trabajo a través de la optimización de la gestión de incidencias es una empresa SAT. Es decir, llevar el control total sobre todas las incidencias de sus clientes, facilitar a sus clientes el poder acceder a la aplicación para que consulten o ingresen nuevos registros y, sobre todo, poder satisfacer todas las necesidades de los técnicos.

En definitiva, la finalidad de la herramienta es proporcionar la ayuda necesaria para que la productividad del trabajo sea favorable para la empresa.

1.3.2 Objetivos específicos

El objetivo principal de este Trabajo Final de Grado es profundizar y adquirir conocimiento sobre el desarrollo de software que implica el análisis, el diseño y la implementación.

Además, poner en práctica el conocimiento adquirido durante toda la carrera desde la planificación, metodologías, uso de patrones, etc. Así como trabajar en nuevas y sofisticadas tecnologías en el desarrollo web.

Las herramientas para conseguir los objetivos deseados son los siguientes:

- Framework *Laravel* de PHP
- *MySQL*.
- Framework *Bootstrap*, y el uso de *Ajax* que harán que la usabilidad del usuario sea más cómoda, ligera y agradable.

1.4 Metodología

La metodología utilizada será el Modelo incremental básico o desarrollo incremental. Esta metodología se caracteriza por proporcionar funcionalidades básicas de manera rápida como, por ejemplo, el sistema de autenticación de usuarios o funcionalidades de la aplicación. También, conlleva a un bajo riesgo de errores.

Seguidamente, se detalla en la *figura 1* el modelo incremental en el tiempo del proyecto.

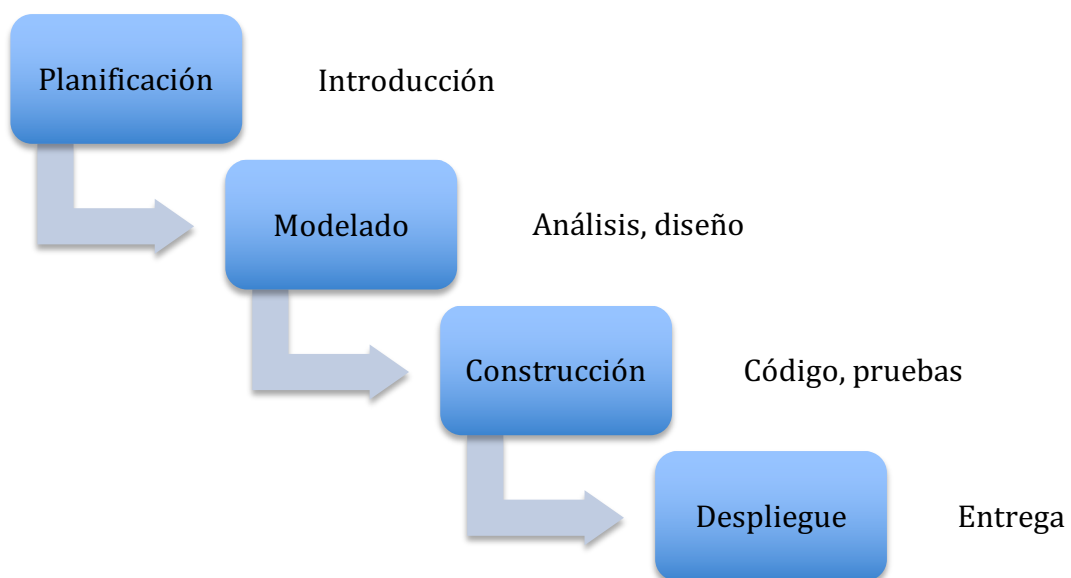
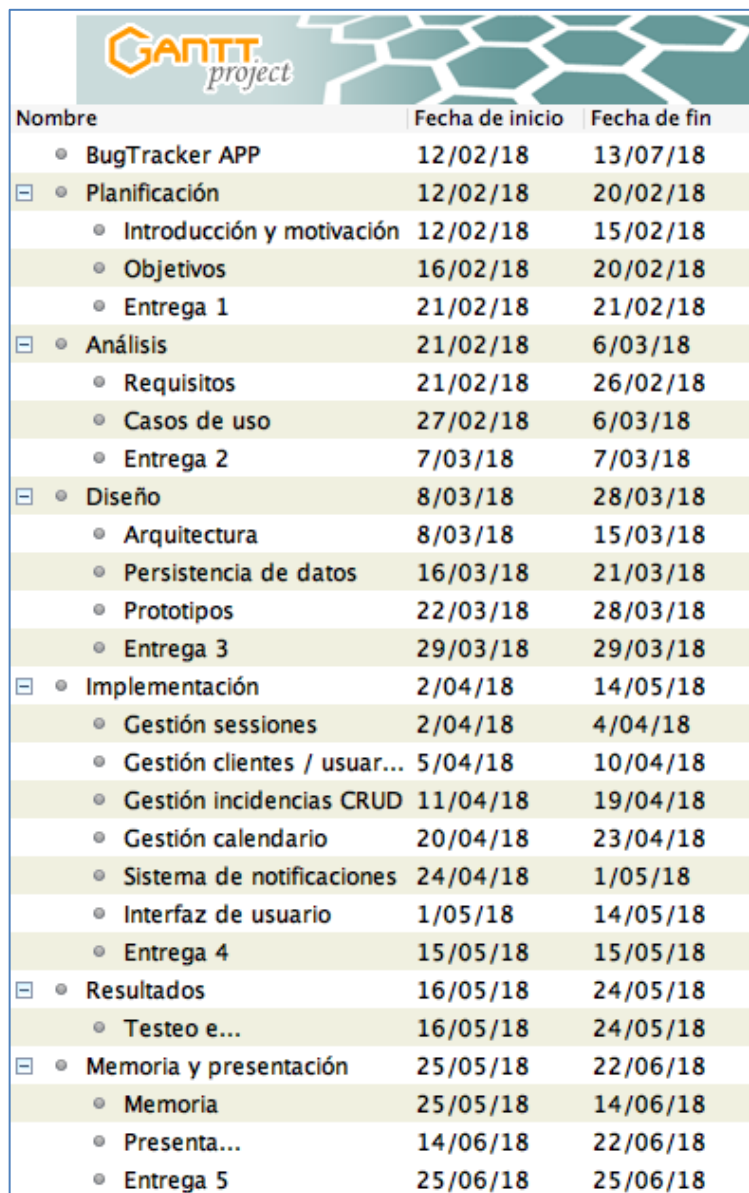


Figura 1: Modelo incremental

1.5 Planificación temporal

1.5.1 Distribución del tiempo

En la *figura 2* se muestra la planificación que se llevará a cabo en este proyecto, así como las entregas.



Nombre	Fecha de inicio	Fecha de fin
• BugTracker APP	12/02/18	13/07/18
[-] • Planificación	12/02/18	20/02/18
• Introducción y motivación	12/02/18	15/02/18
• Objetivos	16/02/18	20/02/18
• Entrega 1	21/02/18	21/02/18
[-] • Análisis	21/02/18	6/03/18
• Requisitos	21/02/18	26/02/18
• Casos de uso	27/02/18	6/03/18
• Entrega 2	7/03/18	7/03/18
[-] • Diseño	8/03/18	28/03/18
• Arquitectura	8/03/18	15/03/18
• Persistencia de datos	16/03/18	21/03/18
• Prototipos	22/03/18	28/03/18
• Entrega 3	29/03/18	29/03/18
[-] • Implementación	2/04/18	14/05/18
• Gestión sesiones	2/04/18	4/04/18
• Gestión clientes / usuar...	5/04/18	10/04/18
• Gestión incidencias CRUD	11/04/18	19/04/18
• Gestión calendario	20/04/18	23/04/18
• Sistema de notificaciones	24/04/18	1/05/18
• Interfaz de usuario	1/05/18	14/05/18
• Entrega 4	15/05/18	15/05/18
[-] • Resultados	16/05/18	24/05/18
• Testeo e...	16/05/18	24/05/18
[-] • Memoria y presentación	25/05/18	22/06/18
• Memoria	25/05/18	14/06/18
• Presenta...	14/06/18	22/06/18
• Entrega 5	25/06/18	25/06/18

Figura 2: Hitos y distribución temporal del proyecto

1.5.2 Diagrama de Gantt

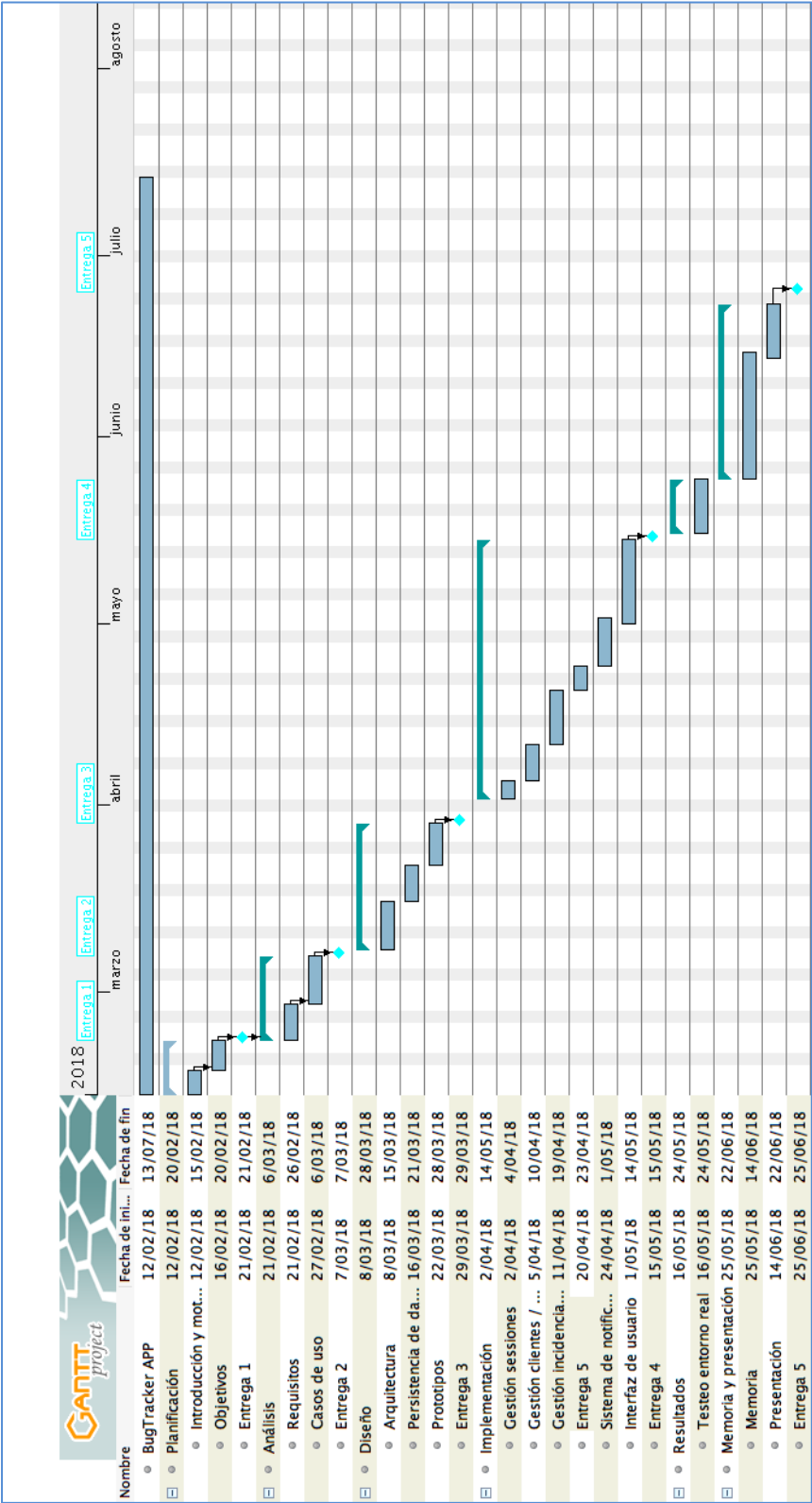


Figura 3: Planificación del proyecto

2. Análisis del sistema

2.1 Requisitos iniciales

En los siguientes apartados se describirán en el escenario, las funcionalidades a desarrollar en esta aplicación y todas las necesidades del cliente.

2.1.1 Requisitos funcionales

Seguidamente se detallarán todas las funcionalidades agrupadas por módulos y los diferentes roles en el sistema.

Gestión de incidencias

- Crear, modificar o borrar incidencias a través del área privada de los agentes o desde el área privada de los clientes.
- Poder seleccionar el tipo de prioridad, reproducibilidad y de otros datos que puedan ayudar a los agentes.
- Poder asignar las incidencias a los agentes. El administrador podrá asignar una o más incidencias y/o los agentes serán capaces de autoasignárselas.
- También, se tiene que poder crear incidencias que sean del tipo evento para poder gestionar la programación diaria de los agentes.
- En una incidencia se podrá añadir archivos adjuntos y/o comentarios, de esta manera, mantendremos una comunicación directa con el cliente.
- En una incidencia se podrán añadir varias actuaciones, así quedará un registro de todas las actuaciones que se han llevado a cabo.
- Una vez se solucione una incidencia, el sistema será capaz de notificar al cliente con un mensaje que le llegará a su panel de control y, también, vía correo electrónico.
- Poder ver un listado diario de todas las incidencias asignadas a un agente, las incidencias que no se han cumplido o las incidencias sin asignación.
- Poder filtrar por fecha, por cliente, por la prioridad o demás características.
- Poder crear categorías para gestionar su propia área de trabajo de cada agente.

Gestión de clientes

- Crear, modificar o borrar clientes, añadir sus datos principales y su ubicación.

- Poder ver un listado de todos los clientes activos o no.
- Estos tendrán un panel de administración para que puedan crear incidencias.

Gestión de calendarios

- Crear, modificar o borrar un evento.
- Poder asignar eventos.
- Añadir notificaciones si se le asigna a un trabajador o a un cliente.
- Tener compartido el calendario con todos los trabajadores.
- Poder ver como administrador todos los eventos programados de mi equipo y, como agente, poder ver mi agenda programada.

Gestión de Usuarios

- Poder crear, modificar o borrar usuarios.
- Poder seleccionar el tipo de usuario.
- Ver un listado de los agentes.

Gestión de ajustes

- Poder gestionar las notificaciones de los usuarios, es decir, habilitar y deshabilitar tanto las notificaciones que llegan en la aplicación como la comunicación vía email.
- Poder gestionar el perfil del usuario, pudiendo cambiar el nombre o contraseña.

Roles / Actores:

- **Administrador:** persona con acceso completo a todas las funcionalidades de la aplicación. Capaz de crear usuarios, clientes, asignar tareas, programar calendarios y monitorizar al equipo.
- **Operario/agente:** persona con acceso a todo lo relacionado con las incidencias, ver o programar su calendario, ver los usuarios registrados en la aplicación y consultar las fichas de los clientes.
- **Cliente:** persona con acceso al apartado de incidencias. También podrá acceder a su perfil, donde podrá modificar sus datos y ajustar las notificaciones del sistema.

2.1.2 Requisitos no funcionales

Los requisitos no funcionales son aquellas funciones que no se consideran en los requisitos fundamentales de la aplicación, tales como la seguridad de la aplicación o la usabilidad y diseño de esta. Seguidamente se detallará los requisitos no funcionales que se tendrán en cuenta:

Requisitos de seguridad

- **El acceso a la aplicación:** sólo será exclusivo para los agentes de la empresa y los clientes. Por ello el sistema se dotará de un sistema de autenticación con un usuario y contraseña únicos.
- **Inyecciones SQL:** ya que la aplicación hace uso de datos sensibles, como sería el caso de los datos de los clientes o las propias incidencias. Otra preventiva que se usará es la de sanear los inputs de los formularios.
- **Ataques *Cross site scripting (XXS)*:** es importante tener en cuenta estos ataques para así prevenir que terceras personas puedan robar información sensible, secuestrar sesiones de usuarios o subyugar la integridad del sistema.

Requisitos de interfaz

Interfaz diseñada para vista escritorio:

- Las diferentes interfaces de la aplicación web se dotarán con un estilo minimalista y con diseños amigables que harán que el usuario pueda acceder a todo el contenido de manera fácil e intuitiva.
- Se usará la tecnología **Ajax** para mostrar, consultar y grabar datos de manera ágil y asíncrona, sin que el usuario tenga que esperar a que se carguen las pantallas.
- Se usará la vista adaptativa para cualquier dimensión de pantalla, **responsive web design**.

2.2 Casos de uso

Una vez hayamos definido los actores y los requerimientos funcionales y no funcionales de nuestra aplicación vamos a desarrollar los diferentes casos de usos del sistema.

2.2.1 Diagrama de casos de uso

Sistema de autenticación

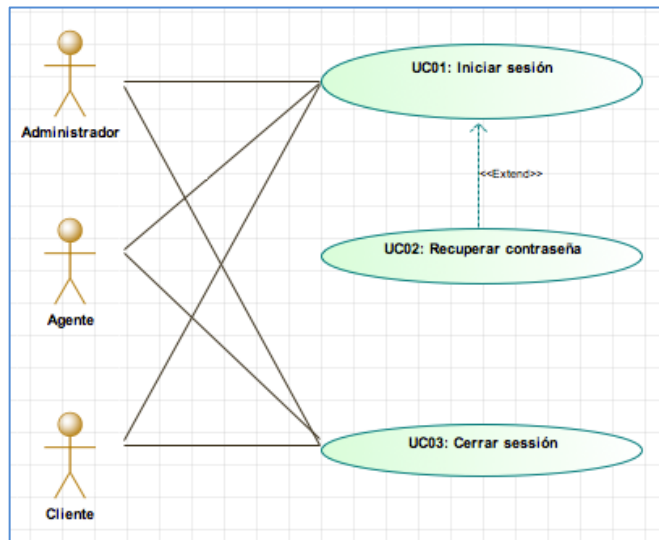


Figura 4: Diagrama de caso de uso - Autenticación

Sistema de Incidencias

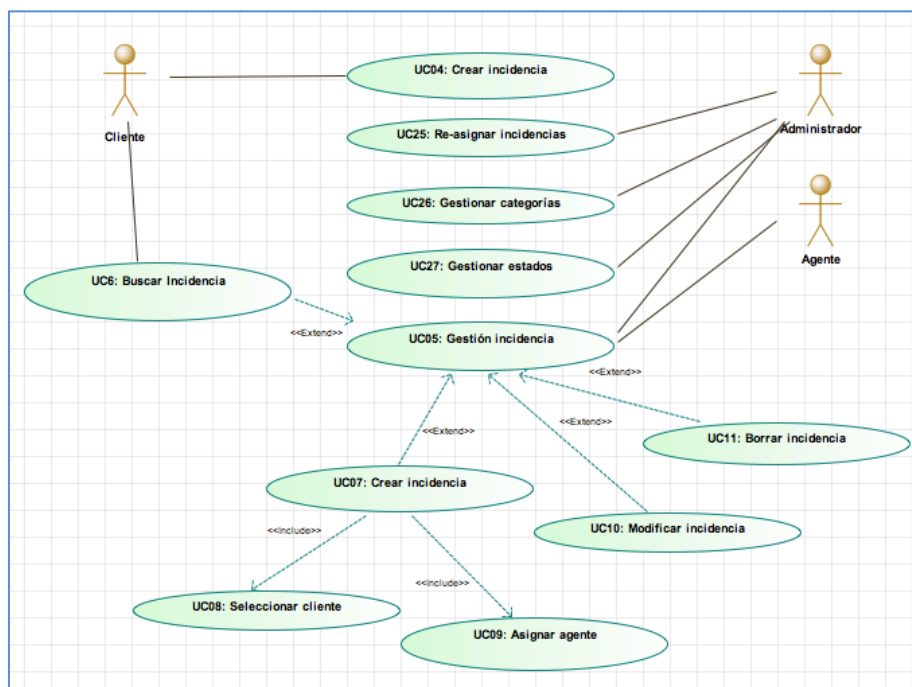


Figura 5: Diagrama de caso de uso - Incidencias

Sistema de Clientes

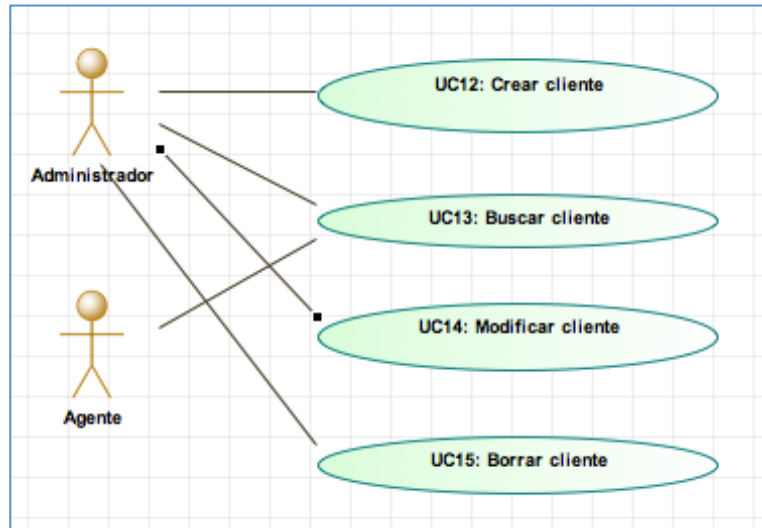


Figura 6: Diagrama de caso de uso - Clientes

Sistema de Calendarios

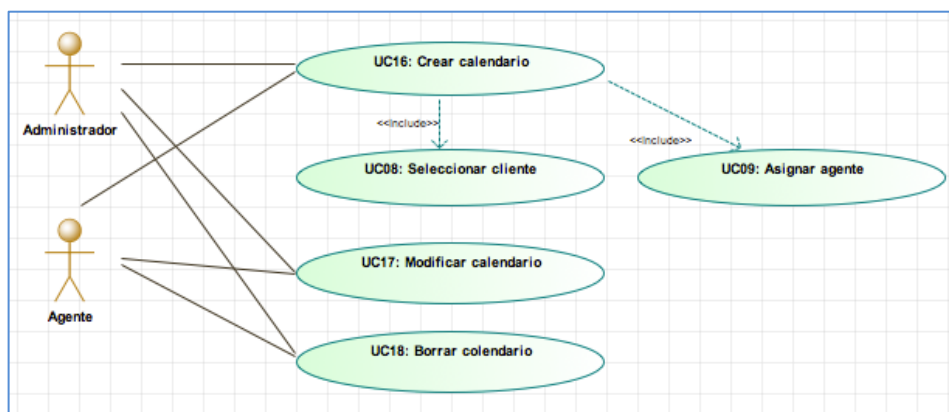


Figura 7: Diagrama de caso de uso - Calendarios

Sistema de Usuarios

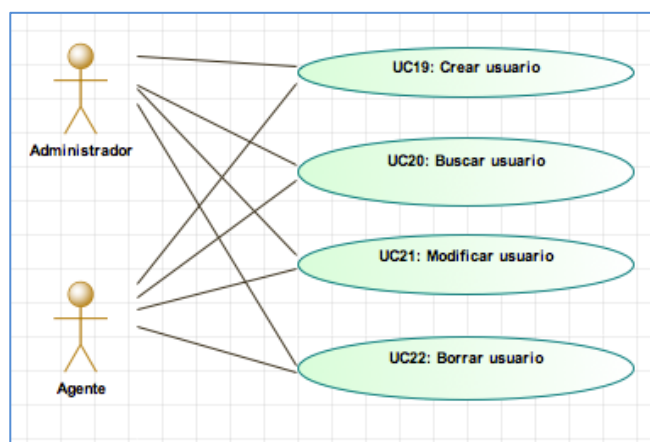


Figura 8: Diagrama de caso de uso - Usuarios

Sistema de Ajustes

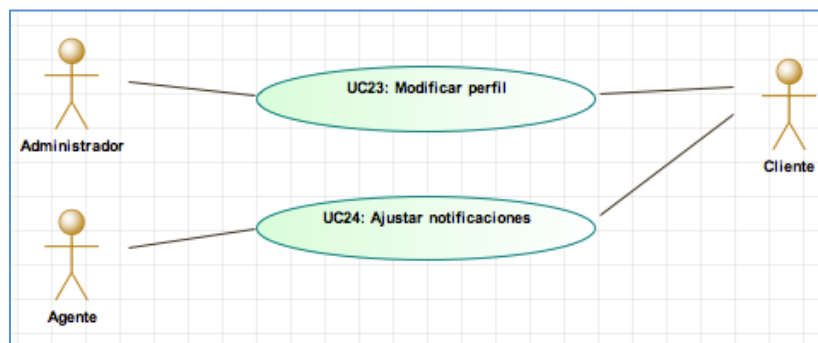


Figura 9: Diagrama de caso de uso - Ajustes

2.2.2 Descripción textual de casos de uso

Seguidamente se detallará los casos de uso textuales de la aplicación.

UC01: Iniciar sesión	
Descripción:	Permite a los actores iniciar sesión en el sistema.
Actores:	Administrador, agentes y clientes.
Pre-condición:	<ul style="list-style-type: none">Los actores han de acceder a la página principalTienen que estar registrados en el sistema, mediante UC19: crear usuario
Flujo principal:	<ol style="list-style-type: none">El actor accede a la página principal de inicio de sesiónEl sistema muestra un formulario pidiendo el nombre de usuario o e-mail y una contraseñaEl actor introduce los datos y hace clic en Entrar.El sistema valida los datos y muestra el panel principal de cada actor.
Flujo alternativo	4a. Si los datos no son válidos no puede entrar en el sistema.
Post-condición:	El sistema comprueba los datos y muestra para cada actor su panel principal.

UC02: Recuperar contraseña	
Descripción:	Permite a los actores recuperar su contraseña en el sistema
Actores:	Administrador, agentes y clientes.
Pre-condición:	<ul style="list-style-type: none">Los actores han de acceder a la página principalLos actores tienen que estar registrados en el sistema, mediante UC19: crear usuario

Flujo principal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción de recuperar contraseña 2. El sistema pedirá el email del actor 3. El actor introduce su email 4. El sistema enviará un email con un link para introducir la nueva contraseña.
Post-condición:	El sistema envía un email con un link para restablecer la contraseña y muestra la vista del UC01.

UC03: Cerrar sesión	
Descripción:	Permite a los actores cerrar la sesión en la aplicación
Actores:	Administrador, agentes y clientes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El actor desde cualquier parte del sistema, en la parte superior selecciona la opción <i>Logout</i> 2. El sistema cierra la sesión del actor
Post-condición:	El sistema redirige automáticamente a la página principal.

UC04: Crear incidencia	
Descripción:	Permite crear incidencias.
Actores:	Clientes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra en la parte izquierda la opción de Incidencias 2. El actor seleccionara la opción Crear incidencia 3. El sistema mostrará un pop-up con un formulario para crear la incidencia. <ol style="list-style-type: none"> a) El sistema muestra dos campos requeridos como el título y la descripción de la incidencia, también muestra como opcionales las opciones de añadir sistema operativo, la reproducibilidad de la incidencia y la prioridad. b) El sistema mostrará una opción para categorizar la incidencia c) El sistema mostrará la opción de UC04: Crear categorías d) El sistema mostrará la opción de añadirlo a un calendario

	4. El actor rellena los datos y hace clic en Crear.
Post-condición:	Una vez guardada la incidencia, el sistema enviará una notificación y volverá al listado de incidencias.

UC5: Gestionar incidencia	
Descripción:	Permite gestionar las incidencias.
Actores:	Administradores y los agentes
Pre-condición:	<ul style="list-style-type: none"> Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> El actor selecciona la opción de Incidencias, en el menú lateral El sistema muestra todas las incidencias que se han creado y las ordena por fecha descendiente. El sistema muestra opción filtrar incidencias, crear, editar, borrar
Post-condición:	-

UC06: Buscar incidencia	
Descripción:	Permite buscar y filtrar incidencias
Actores:	Administrador, agentes y clientes
Pre-condición:	<ul style="list-style-type: none"> Los actores han de haber realizado el UC01: Iniciar sesión Haber realizado alguna UC07: Crear incidencia
Flujo principal:	<ol style="list-style-type: none"> El sistema por defecto mostrará un listado de todas las incidencias y la opción de buscar o filtrar incidencias. El actor selecciona en la parte superior la opción de buscar y inserta el parámetro de búsqueda El sistema devuelve los registros según el parámetro de búsqueda y vuelve al punto 1. El actor selecciona la opción de filtrar. El sistema mostrará un pequeño formulario con los filtros de búsqueda. El actor filtra El sistema devuelve los registros según el filtrado.
Post-condición:	El sistema comprueba que existen dichos parámetros de búsqueda y devuelve los registros.

UC07: Crear incidencia	
Descripción:	Permite crear incidencias.
Actores:	Administrador y agentes
Pre-condición:	<ul style="list-style-type: none"> Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> El sistema muestra en la parte izquierda la opción de Incidencias El actor selecciona la opción El sistema muestra la opción de Crear incidencia El actor seleccionara la opción Crear incidencia El sistema mostrará un pop-up con un formulario para crear la incidencia. <ol style="list-style-type: none"> El sistema muestra dos campos requeridos como el título y la descripción de la incidencia, también muestra como opcionales las opciones de añadir teléfono, sistema operativo, la reproducibilidad de la incidencia, la prioridad y el estado de la incidencia. El sistema mostrará la opción de asignar a un agente El sistema mostrará una opción para categorizar la incidencia El sistema mostrará la opción de añadirlo a un calendario El actor rellena los datos y hace clic en Crear.
Post-condición:	Una vez guardada la incidencia, el sistema enviará una notificación y volverá al listado de incidencias.

UC08: Seleccionar cliente	
Descripción:	Permite asignar la incidencia a un cliente
Actores:	Administrador y agentes
Pre-condición:	<ul style="list-style-type: none"> Los actores han de haber realizado el UC01: Iniciar sesión Solo lo pueden hacer los administradores y agentes
Flujo principal:	<ol style="list-style-type: none"> El sistema muestra un listado con todas las incidencias El actor selecciona incidencia El sistema abre la incidencia en un pop-up con el formulario El actor selecciona a un cliente en el campo de Cliente y hace clic en guardar.

	7. El sistema devuelve un mensaje de satisfacción
Post-condición:	El sistema guarda la incidencia y añade en el histórico del cliente

UC09: Asignar agente	
Descripción:	Permite asignar una incidencia a un agente.
Actores:	Administrador y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado alguna UC05: Crear incidencia
Flujo principal:	5) El sistema muestra un listado con todas las incidencias 6) El actor selecciona incidencia 7) El sistema abre la incidencia en un pop-up con el formulario 8) El actor selecciona a un usuario en el campo de Asignar a y hace clic en guardar. 9) El sistema devuelve un mensaje de satisfacción
Post-condición:	El sistema asigna la incidencia, y notifica al agente que se le ha asignado la incidencia.

UC10: Modificar incidencia	
Descripción:	Permite modificar, actualizar las incidencias
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado alguna UC05: Crear incidencia
Flujo principal:	1. El sistema muestra un listado con todas las incidencias. 2. El actor selecciona una de ellas. 3. El sistema muestra la información completa de la incidencia en forma de pop-up. 4. El actor actualiza los datos de la incidencia y hace clic en Guardar. 5. El sistema guarda los datos y devuelve un mensaje de satisfacción.
Post-condición:	El sistema comprueba que los datos requeridos se cumplan, de lo contrario no guarda la incidencia.

	Si se cumplen el sistema guarda la modificación y si es necesario notifica al cliente.
--	--

UC11: Borrar incidencia	
Descripción:	Permite borrar una incidencia
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado alguna UC05: Crear incidencia
Flujo principal:	<ol style="list-style-type: none"> 1) Los actores seleccionan una incidencia 2) El sistema muestra un pop-up con la información de la incidencia y un botón de Borrar. 3) Los actores hacen clic en Borrar 4) El sistema devuelve un mensaje de satisfacción.
Post-condición:	El sistema borra la incidencia del sistema y actualiza el listado.

UC12: Crear cliente	
Descripción:	Permite crear clientes dentro de la aplicación.
Actores:	Administradores
Pre-condición:	Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra en la parte izquierda la opción de Clientes 2. El actor seleccionará la opción Crear Cliente 3. El sistema muestra un pop-up con un formulario pidiendo los siguientes datos: <ol style="list-style-type: none"> a. Nombre b. Descripción c. Tipo de suscripción d. Localización e. Y si está activo o no 4. El actor rellena los datos y hace clic en Guardar. 5. El sistema guarda los datos y devuelve un mensaje de satisfacción.
Post-condición:	El sistema guarda el nuevo cliente

UC13: Buscar cliente	
Descripción:	Permite buscar un cliente
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado algún UC11: Crear cliente
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra la opción de Cliente en el menú izquierdo 2. El actor selecciona dicha opción. 3. El sistema devuelve el listado de clientes con la opción de Buscar cliente. 4. El actor escribe un parámetro de búsqueda y hace clic en buscar. 5. El sistema devuelve los registros que se asemejan a la búsqueda.
Post-condición:	El sistema comprueba y devuelve el registro en el listado de clientes.

UC14: Modificar cliente	
Descripción:	Permite modificar los datos de los clientes
Actores:	Administradores
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado algún UC11: Crear cliente
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra un listado con todas los clientes. 2. El actor selecciona un cliente 3. El sistema muestra la información completa del cliente en forma de pop-up. 4. El actor actualiza los datos del cliente y hace clic en Guardar. 5. El sistema guarda los datos y devuelve un mensaje de satisfacción.
Post-condición:	<p>El sistema comprueba que los datos requeridos se cumplan, de lo contrario no guarda el cliente.</p> <p>Si se cumplen el sistema guarda la modificación.</p>

UC15: Borrar cliente	
Descripción:	Permite borrar un cliente

Actores:	Administradores
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado alguna UC11: Crear cliente
Flujo principal:	<ol style="list-style-type: none"> 1) El sistema muestra el listado de clientes con una opción a la derecha de borrar. 2) El actor hace clic en borrar. 3) El sistema devuelve un mensaje de satisfacción.
Post-condición:	El sistema borra el cliente del sistema y actualiza el listado.

UC16: Crear calendario	
Descripción:	Permite crear nuevos eventos de incidencias
Actores:	Administradores y agentes
Pre-condición:	Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra en la parte izquierda la opción de Calendario y también muestra una pequeña opción al realizar el UC05: Crear incidencia. 2. El actor seleccionará la opción Crear nuevo evento 3. El sistema muestra un pop-up con un formulario pidiendo los siguientes datos: <ol style="list-style-type: none"> a. Título b. Descripción c. A quien le asigna el evento d. Fecha de inicio y fecha fin 4. El actor rellena los datos y hace clic en Guardar. 5. El sistema guarda los y devuelve un mensaje de satisfacción.
Post-condición:	El sistema guarda el nuevo evento y notifica al agente que se le ha asignado el evento.

UC17: Modificar calendario	
Descripción:	Permite modificar un evento programado
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado algún UC15: Crear calendario

Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra un calendario con todos los eventos programados. 2. El actor selecciona un evento 3. El sistema muestra la información completa del evento en forma de pop-up. 4. El actor actualiza los datos del evento y hace clic en Guardar. 5. El sistema guarda los datos y devuelve un mensaje de satisfacción.
Post-condición:	<p>El sistema comprueba que los datos requeridos se cumplan, de lo contrario no guarda el evento.</p> <p>Si se cumplen el sistema guarda la modificación.</p>

UC18: Borrar calendario	
Descripción:	Permite borrar un evento programado
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado alguna UC15: Crear calendario
Flujo principal:	<ol style="list-style-type: none"> 1) El sistema muestra el calendario de los eventos con una opción a la derecha de borrar. 2) El actor hace clic en borrar. 3) El sistema devuelve un mensaje de satisfacción.
Post-condición:	El sistema borra el evento del sistema y actualiza el calendario.

UC19: Crear usuario	
Descripción:	Permite crear nuevos agentes dentro de la aplicación.
Actores:	Administradores y agentes
Pre-condición:	Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra en la parte izquierda la opción de Usuarios 2. El actor seleccionará la opción Crear usuario 3. El sistema muestra un pop-up con un formulario pidiendo los siguientes datos: <ol style="list-style-type: none"> a. Nombre b. Correo electrónico

	<ul style="list-style-type: none"> c. Teléfono d. Contraseña e. El tipo de rol f. Y si está activo o no <ol style="list-style-type: none"> 4. El actor rellena los datos y hace clic en Guardar. 5. El sistema guarda los datos y devuelve un mensaje de satisfacción.
Post-condición:	<p>El sistema comprueba que los datos requeridos se cumplan, de lo contrario no crea el nuevo usuario.</p> <p>Si se cumplen el sistema lo crea.</p>

UC20: Buscar usuario	
Descripción:	Permite buscar un agente
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado algún UC19: Crear usuario
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra la opción de Usuarios en el menú izquierdo 2. El actor selecciona dicha opción. 3. El sistema muestra un listado con todos los agentes activos y la opción de Buscar agente. 4. El actor escribe un parámetro de búsqueda y hace clic en buscar. 5. El sistema devuelve los registros que se asemejan a la búsqueda, en formato de lista.
Post-condición:	El sistema comprueba y devuelve el registro en el listado de agentes.

UC21: Modificar usuario	
Descripción:	Permite modificar los datos de los clientes
Actores:	Administradores y agentes
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado algún UC19: Crear Usuario
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra un listado con todos los usuarios.

	<ol style="list-style-type: none"> 2. El actor selecciona un agente 3. El sistema muestra la información completa del agente en forma de pop-up. 4. El actor actualiza los datos del agente y hace clic en Guardar. 5. El sistema guarda los datos y devuelve un mensaje de satisfacción.
Post-condición:	<p>El sistema comprueba que los datos requeridos se cumplan, de lo contrario no guarda el agente.</p> <p>Si se cumplen el sistema guarda la modificación.</p>

UC22: Borrar usuario	
Descripción:	Permite borrar un agente
Actores:	Administradores
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado alguna UC19: Crear agente
Flujo principal:	<ol style="list-style-type: none"> 1) El sistema muestra el listado de usuarios con una opción a la derecha de borrar. 2) El actor hace clic en borrar. 3) El sistema devuelve un mensaje de satisfacción.
Post-condición:	El sistema borra el agente del sistema y actualiza el listado.

UC23: Modificar datos	
Descripción:	Permite a los usuarios del sistema actualizar sus datos básicos.
Actores:	Administrador, agentes y clientes
Pre-condición:	Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra la opción de Editar perfil en el menú izquierdo 2. El actor selecciona dicha opción. 3. El sistema muestra toda la información del usuario en un formulario 4. El actor modifica sus datos y hace clic en Actualizar 5. El sistema guarda la nueva modificación y muestra un mensaje de satisfacción

Post-condición:	El sistema guarda la nueva modificación del usuario.

UC24: Ajustar notificaciones	
Descripción:	Permite a los usuarios de la aplicación poder habilitar o deshabilitar las notificaciones de escritorio
Actores:	Administradores, agentes y clientes
Pre-condición:	Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra la opción de Ajustes en el menú izquierdo 2. El actor selecciona dicha opción. 3. El sistema muestra un input de habilitar / deshabilitar notificaciones 4. El usuario selecciona alguna de las opciones 5. El sistema guarda la nueva configuración
Post-condición:	El sistema guarda la configuración del actor.

UC25: Reasignar incidencias	
Descripción:	Permite a los administradores el volcado de todas o una de las incidencias a otro agente.
Actores:	Administrador
Pre-condición:	<ul style="list-style-type: none"> • Los actores han de haber realizado el UC01: Iniciar sesión • Haber realizado algún UC05: Crear incidencia • Haber realizado algún UC19: Crear usuario
Flujo principal:	<ol style="list-style-type: none"> 1. El sistema muestra la opción de Ajustes en el menú izquierdo 2. El actor selecciona dicha opción. 3. El sistema muestra un apartado de Volcado de incidencias 4. El actor selecciona un agente 5. El sistema devuelve todas las incidencias de dicho agente y un listado de agentes al que queremos volcar las incidencias 6. El actor selecciona una o todas las incidencias, y posteriormente selecciona al agente al que queremos volcar las incidencias

	7. El sistema asigna dichas asignaciones al nuevo agente y muestra un mensaje de satisfacción
Post-condición:	El sistema guarda las asignaciones, y notifica al nuevo agente.

UC26: Gestionar categorías	
Descripción:	Permite gestionar categorías para las incidencias.
Actores:	Administradores
Pre-condición:	<ul style="list-style-type: none"> Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	4. El actor selecciona la opción de Ajuste de incidencias 5. El sistema muestra en la parte derecha un listado con las categorías 6. El sistema muestra opción de añadir o modificar una categoría
Post-condición:	-

UC27: Gestionar estados	
Descripción:	Permite gestionar estados para las incidencias.
Actores:	Administradores
Pre-condición:	<ul style="list-style-type: none"> Los actores han de haber realizado el UC01: Iniciar sesión
Flujo principal:	7. El actor selecciona la opción de Ajuste de incidencias 8. El sistema muestra en la parte derecha un listado con los estados 9. El sistema muestra opción de añadir o modificar una categoría
Post-condición:	-

3. Diseño

3.1 Decisiones tecnológicas

Considerando todos los requisitos, funcionales y no funcionales, y teniendo en cuenta los dispositivos en los que se ejecutará la aplicación, las decisiones tecnológicas para el desarrollo de la aplicación web son las siguientes:

Después de analizar los diferentes tipos de lenguajes de programación y los requisitos funcionales de la aplicación, se ha decidido utilizar el framework *Laravel* 5.6 hecho en PHP. Una de las decisiones principales es que es código abierto, lleva muchos años como lenguaje para desarrollar aplicaciones web, existen muchos componentes y la puesta en marcha en un entorno real es muy fácil. También, nos permite crear la aplicación de forma sencilla y ordenada siguiendo el patrón MVC y nos ofrece seguridad contra posibles ataques.

Otra característica importante del framework es que dispone de su propio ORM, que crea objetos virtuales sobre la base de datos relacional y hace que la interacción con la base de datos sea orientada a objetos. Es modulable y con un amplio sistema de paquetes como por ejemplo la autenticación. Y por último el uso de middlewares, que nos permitirán filtrar las peticiones HTTP.

En cuanto el diseño de la interfaz de usuario se utilizará el framework *Bootstrap* 4 ya que podemos re-utilizar los componentes *HTML* y *CSS* que nos ofrece.

Para darle dinamismo a la aplicación se usará el lenguaje de programación JavaScript junto a la librería *jQuery* y el uso de la tecnología *AJAX*.

Para la persistencia de datos se usará *Structured Query Language* (SQL) y el motor de base de datos *MySQL*.

3.2 Arquitectura del software

La arquitectura de la aplicación se base en el patrón Modelo-Vista-Controlador (MVC) que separa la lógica de control, los datos de la aplicación y la interfaz de usuario.

En la *figura 10* se muestra un ejemplo del flujo que se realiza cuando el usuario hace una petición *HTTP*.

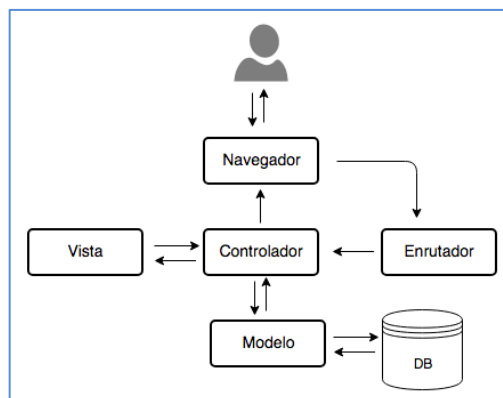


Figura 10: Petición HTTP

A continuación, explicaremos en detalle estos 3 componentes.

3.2.1 Modelo

Antes de explicar que es un Modelo, se hará un pequeño inciso sobre la manera de trabajar con la persistencia de datos en *Laravel*.

Las migraciones es un sistema de control de versión de la base de datos. Lo utilizamos para definir los campos de una tabla o en alguna modificación en el futuro.

Por ejemplo, para crear una nueva tabla usaremos la siguiente línea de comando, gracias a la herramienta **Artisan**:

```
php artisan make:migrate create_bug_table --create=bugs
```

Este comando generará un fichero PHP con dos métodos: el método *up()* es el que se encargará de crear los campos de la tabla y el método *down()* que sirve para eliminar los campos de una tabla o toda una tabla.

Para poder ejecutar la migración basta con escribir la siguiente línea de comando:

```
php artisan make:migrate
```

Este comando ejecutará la migración y creará las tablas y/o modificaciones en la base de datos.

La interacción con la base de datos es muy sencilla utilizando la herramienta **Eloquent ORM** (Mapeo Objeto-Relacional), una técnica de programación que nos permite unificar los datos de programación de objetos con una base de datos relacional.

Cada tabla de nuestra aplicación corresponde a un modelo que se guardan como clases en PHP.

3.2.2 Vista

Las vistas son la parte que el usuario verá de nuestra aplicación, generalmente son HTML, CSS, JavaScript y código PHP para mostrar información de nuestra aplicación.

Para tener una estructura ordenada con las vistas utilizaremos el motor de plantillas o *Templates*, que permite modularizar segmentos de código para que estos no se repitan en cada vista.

Utilizaremos la librería **Blade**, que es el motor de plantillas que nos permite mezclar de manera limpia estructuras de control y variables de PHP.

Estas plantillas tienen que tener la extensión *.blade.php. El código que utilizará *Blade* empezará por los símbolos @ o {{.

Con *Blade* podemos mostrar datos y programar estructuras de control, tales como condiciones. También, podemos ir recorriendo el *Array* de datos con la utilización de bucles del tipo *for*, *while* o *foreach*. Otra característica es que *Blade* se encarga de escapar el resultado llamando a **htmlentities** para prevenir errores y sobre todo ataques del tipo **Cross-site scripting** (XSS). Si, por lo contrario, no queremos escapar los datos tendríamos que llamar así a las variables: `{{!! $miVariable !!}}`

A continuación, se describirá cada vista de la aplicación:

Inicio de sesión

- Página que permitirá a los usuarios entrar dentro de la aplicación.

Dashboard - Página principal

- Página que mostrará de forma rápida las incidencias nuevas, activas, las que están sin asignar y las urgentes.
- Un **widget** que mostrará los eventos en forma de lista del usuario de la aplicación.
- Otro **widget** donde mostrará la información de las actividades de los agentes y por último un Time-line con todo el histórico de estados que pasa una incidencia.

Gestión de incidencias

- **Listado de incidencias:** Página donde se mostrará un buscador avanzado y un listado completo de todas las incidencias del sistema (estos diferentes tipos tendrán una vista parcial cada una). También, se encuentra la opción de crear una nueva incidencia. A este apartado tendrán acceso todos los usuarios registrados, pero dependiendo del rol, se ocultará o mostrarán algunas opciones.

- **Modificar / ver incidencia:** Página del tipo vista parcial (*partial-view*) donde se mostrará en forma de modal con la información de una incidencia.
- **Ajuste de incidencias:** Página de ajustes de incidencias donde el Administrador podrá re-assignar incidencias, crear categorías y/o estados de una incidencia.

Gestión de calendarios

- **Listado de eventos:** Página donde se mostrará un calendario con los eventos programados por los usuarios.
- **Modificar/ver evento:** Página del tipo vista parcial donde se mostrará en forma de modal toda la información del evento.

Gestión de clientes

- **Listado de clientes:** Página donde se mostrará en forma de lista todos los clientes de la aplicación.
- **Modificar/ver cliente:** Página del tipo vista parcial donde se mostrará en forma de modal toda la información del cliente.

Gestión de usuarios

- **Listado de usuarios:** Página donde se mostrará en forma de lista todos los usuarios de la aplicación.
- **Modificar / ver usuario:** Página del tipo vista parcial donde se mostrará en forma de modal toda la información del usuario.

Gestión de ajustes

- Página donde se mostrará un formulario para poder cambiar la información básica del usuario y, también, las preferencias de habilitar o deshabilitar las notificaciones del sistema.

3.2.3 Controlador

Los controladores sirven para organizar y separar el código y crear clases que nos ayuden a agrupar todos los recursos de una funcionalidad. Es aquí donde se pondrá toda la lógica de una funcionalidad.

Los controladores se almacenan en la carpeta *app/Http/controllers* y debemos añadir el sufijo **Controller** como por ejemplo *BugsController*.

Veamos cómo funciona un controlador y por qué nos hace falta en esta aplicación.

Supongamos que queremos mostrar la información de un Bug en concreto, lo que haríamos es hacer la petición pasándole un ID, por ejemplo. Y, seguidamente, un método para que nos sirva el recurso que estamos solicitando. Este método se encargará de buscar en el modelo *Bug*, y retornar a la vista sus datos.

Para organizar las peticiones que se hacen por HTTP, *Laravel* usa el recurso rutas (*Routes*). A continuación, explicaremos brevemente en qué consiste este recurso.

Rutas

Todas las rutas de la aplicación se definen en el siguiente directorio *routes/web.php*. Como bien lo indica su nombre, esto nos ayudará a enrutar todas las peticiones HTTP que se realicen en nuestra aplicación. Por ejemplo, si el usuario accede a la ruta <http://localhost/admin/bugs> y si lo tenemos definido como una petición **GET**, este nos servirá la petición que estamos realizando. Por lo contrario, nos devolverá un error **HTTP 404** (página no encontrada).

También, podemos añadir parámetros en nuestras rutas. Como por ejemplo, obtener información detallada de un registro, bajo la petición *GET*.

Las rutas trabajan con los siguientes verbos HTTP:

GET	POST	PUT	DELETE	PATCH
-----	------	-----	--------	-------

3.3 Diagrama de clases

El siguiente diagrama se recoge todas las clases y atributos de la aplicación y las relaciones que existen entre ellas.

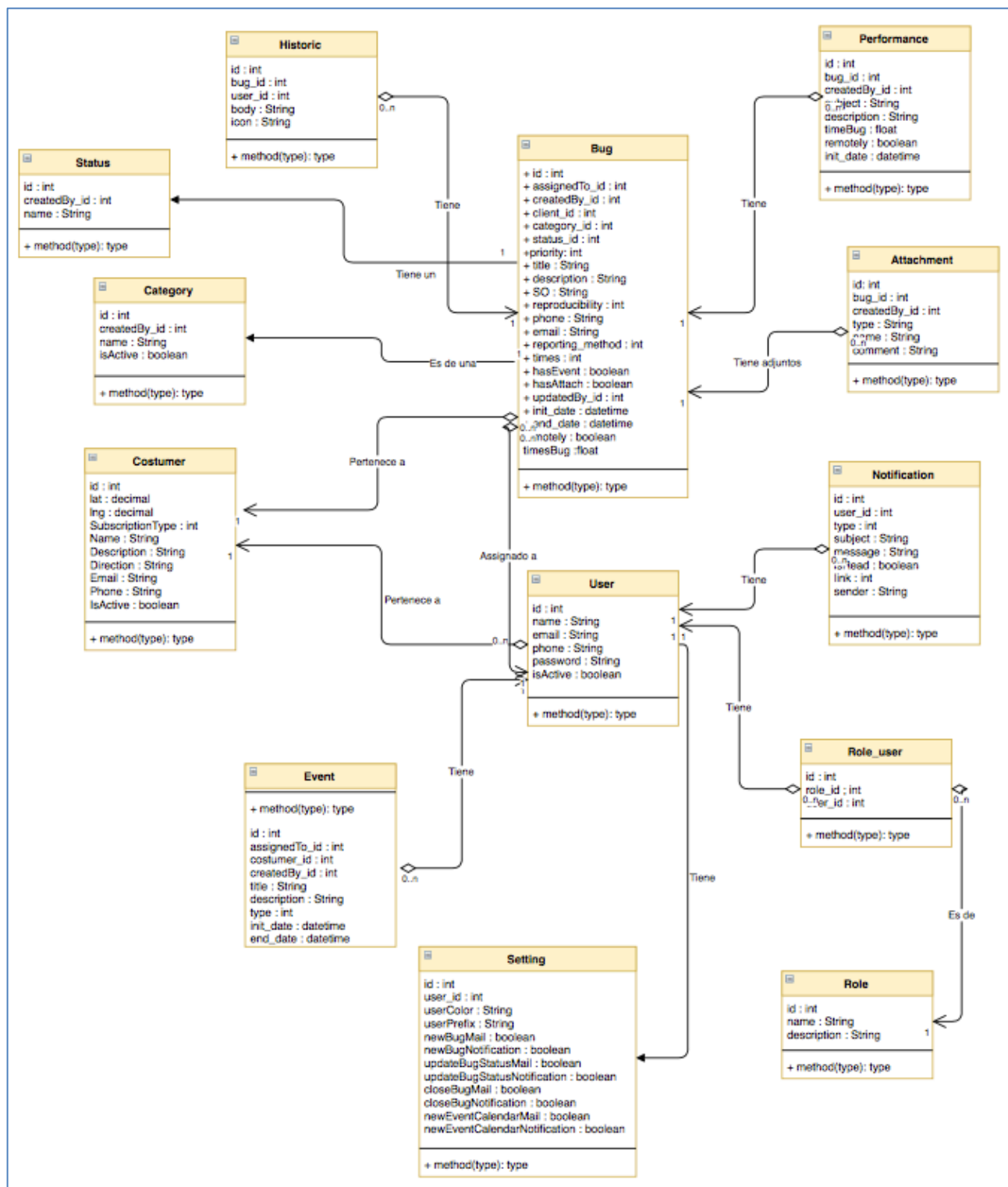


Figura 11: Diagrama de clases de la aplicación

3.4 Prototipos

En el siguiente apartado se mostrarán algunos prototipos de las vistas y los diseños más representativos de la aplicación. Estos diseños no son los definitivos, son solo de carácter informativo. De esta manera, mantenemos al cliente final con una orientación hacia dónde va la aplicación.

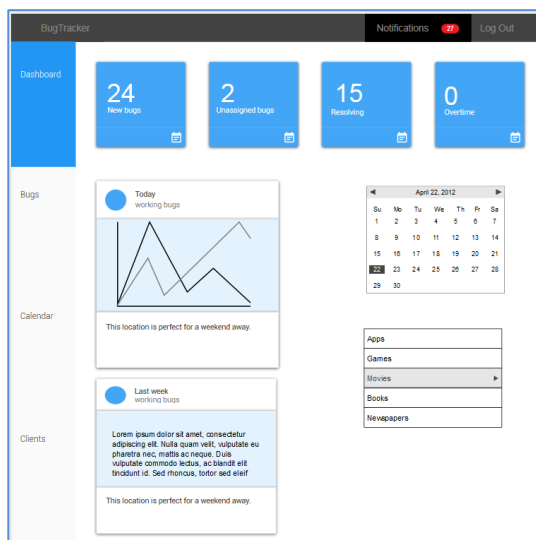


Figura 12: Prototipo del dashboard

En la figura 12, el objetivo es mostrarle al agente una visión general sobre el trabajo diario que debe desempeñar y, también, un recordatorio de los eventos que tenga programado. Este diseño es simple, pero a la vez eficaz ya que nos da la mínima información necesaria.

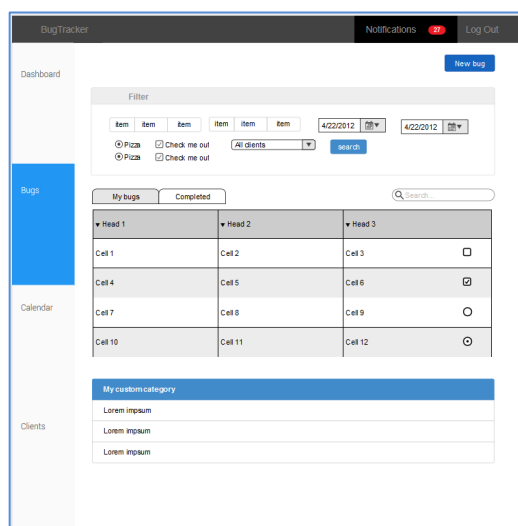


Figura 13: Prototipo de la página de Bugs

En la figura 13, se muestra el prototipo de la pantalla de Bugs, donde se mostrará un filtrado avanzado para buscarlos. Para mostrar todos los Bugs, se añadirán pestañas como, por ejemplo:

mis incidencias, todas las incidencias, las que están sin asignar y las completadas; donde en cada una de ellas aparecerá un listado con diferentes acciones.

Por ejemplo, para los listados de mis incidencias y todas las incidencias, el usuario podrá ver la información básica y podrá completar la incidencia a través de un botón visible en la parte izquierda.

En el caso de que estemos visualizando la pestaña de incidencias sin asignar, aparecerá la información básica y un botón de “Asignármela a mí”. Donde el usuario hará clic y esta aparecerá en el listado de mis incidencias.

También dispondrá, en la parte superior izquierda, un botón para añadir nuevas incidencias.

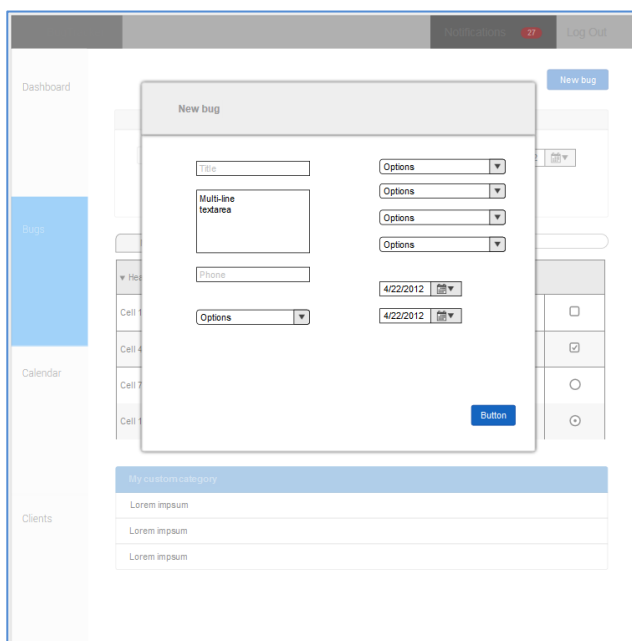
The image shows a web application interface with a sidebar on the left containing links for 'Dashboard', 'Bugs', 'Calendar', and 'Clients'. The main content area features a 'New bug' modal window. This modal contains a 'Title' text input, a 'Multi-line textarea', a 'Phone' text input, and an 'Options' dropdown menu. To the right of these fields are four more 'Options' dropdown menus and two date pickers set to '4/22/2012'. A 'Button' is located at the bottom right of the modal. In the background, a table titled 'My custom category' is visible, containing three rows of placeholder text 'Lorem ipsum'.

Figura 14: Prototipo de visualización de un Bug

En la *figura 14*, se muestra el prototipo de visualización de un bug en concreto. La idea es mostrar los datos dinámicamente, en forma de pop up (modal). También, tendremos pestañas donde en cada una de ellas mostraremos diferentes opciones.

En la primera pestaña se mostrará la información completa de la incidencia; en la siguiente pestaña, se mostrará un mapa con la información del cliente al que se le ha asignado dicha incidencia; en una tercera pestaña se podrán gestionar las actuaciones; y, por último, una cuarta pestaña donde se podrán gestionar las notas o archivos adjuntos.

4. Implementación

Una vez definido el análisis y el diseño, en este apartado, explicaremos todo el desarrollo de los componentes de la aplicación, desde cómo se entra en la aplicación con los distintos roles de la plataforma, pasando por la seguridad en cuanto a la protección de datos hasta las funcionalidades del proyecto.

4.1 El modelo

Relaciones

En esta aplicación es necesario tener varias relaciones entre tablas. Por ejemplo, la relación uno a uno, **hasOne** hace referencia a que un usuario de la aplicación sólo tiene una configuración en el sistema.

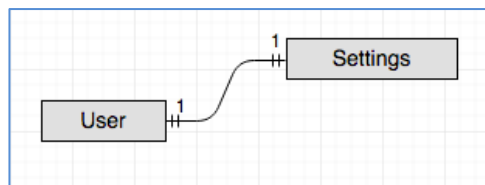


Figura 15: Relación *hasOne*

Otra relación muy usual, es la relación de uno a muchas, **hasMany**, que hace referencia si un *Bug* tiene una o muchas Actuaciones.

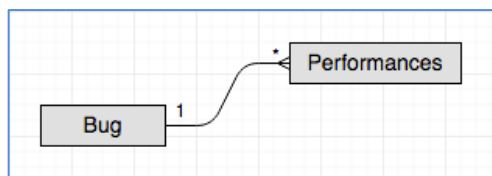


Figura 16: Relación *hasMany*

Una vez se definen las relaciones entre modelos, tenemos que tener en cuenta que se debe añadir la relación inversa, por ejemplo, como hemos visto, un *Bug* puede tener varias Actuaciones, ahora dentro del modelo Actuaciones tenemos que añadir el método **belongsTo**.

```
Public function bug() {  
    Return $this->belongsTo('App\Bug');  
}
```

Figura 17: Ejemplo de la relación inversa

Para obtener datos de las relaciones basta con llamar al método que hace referencia al modelo relacionado y accederemos a sus atributos, por ejemplo, si de un usuario quiero saber si tiene habilitada las notificaciones vía correo, lo haríamos de la siguiente manera:

```
$canNotify = User::find(1)->settings()->newBugMail;
```

Figura 18: obtenemos un Booleano para saber si podemos notificar al usuario 1.

Uso de los modelos

Una vez hemos definido todos los modelos, ya podemos comenzar a trabajar con ellos: crear nuevos registros, modificarlos o borrarlos; y, también, recuperar todos los registros de la base de datos.

Veamos un ejemplo sencillo sobre cómo recuperar todas las incidencias:

Con el método **all()** se pueden recuperar todos los registros de la tabla Bug, este método nos retornara todos los datos en forma de un array. Esto equivale a la consulta *SELECT * FROM bugs*;

```
$allBugs = Bug::all();
```

Figura 19: devuelve todas las incidencias

También, podemos hacer varios filtros para recuperar los datos que nos interesen, por ejemplo, podemos buscar por el estado de una incidencia, ordenarlos, etc.

```
$myCompletedBugs = Bug::where('status_id',5)->get();
```

Figura 20: En este ejemplo hacemos una búsqueda de todos los bugs con el estado igual a 5.

Algunos métodos que podemos realizar a los modelos serían:

All()	Where()	Find()	Max()	Min()
Avg()	Sum()	Get()	Save()	Delete()

4.2 Sistema de autenticación

Como en cualquier plataforma que requiere el manejo de usuarios, es muy importante el sistema de inicio de sesión, ya que es necesario para empezar a trabajar con la aplicación.

El framework nos dota de este componente, pero, en este caso, hemos hecho las siguientes modificaciones:

Autenticación

Como bien hemos comentado antes, hemos hecho la modificación para que solo los usuarios “activos” en la plataforma puedan acceder a ella.

Para ello, hemos sobrescrito el método `credentials()` de los diferentes servicios que nos brinda el framework, a este tipo de servicios se les llama **Facades**. De esta manera, aparte de comprobar las credenciales del usuario, comprobamos si está activo o no.

Una vez el usuario haya pasado la autenticación, en el método `authenticated()` aplicaremos la lógica para redirigir al usuario en función de su perfil, ya sea al panel de administración de los agentes o al panel de administración de los clientes.

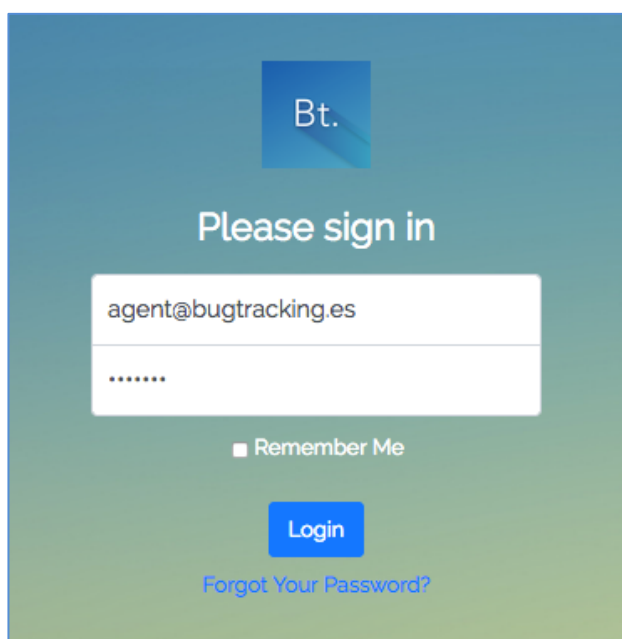


Figura 21: Formulario de acceso a la aplicación

4.3 Middleware

Los middlewares son un servicio que nos ayudará a hacer un filtrado a todas las peticiones HTTP que se realizan en la aplicación. A parte del middleware de autenticación que verifica si el usuario está autenticado en la aplicación, crearemos otras para comprobar tres supuestos casos: el intento de acceder a zonas restringidas, el intento de acceder a recursos que el usuario no tiene permisos y, también, el intento de consultar datos que no pertenezcan a la organización del usuario.

Estos middlewares se pueden implementar a nivel de rutas o de controlador.

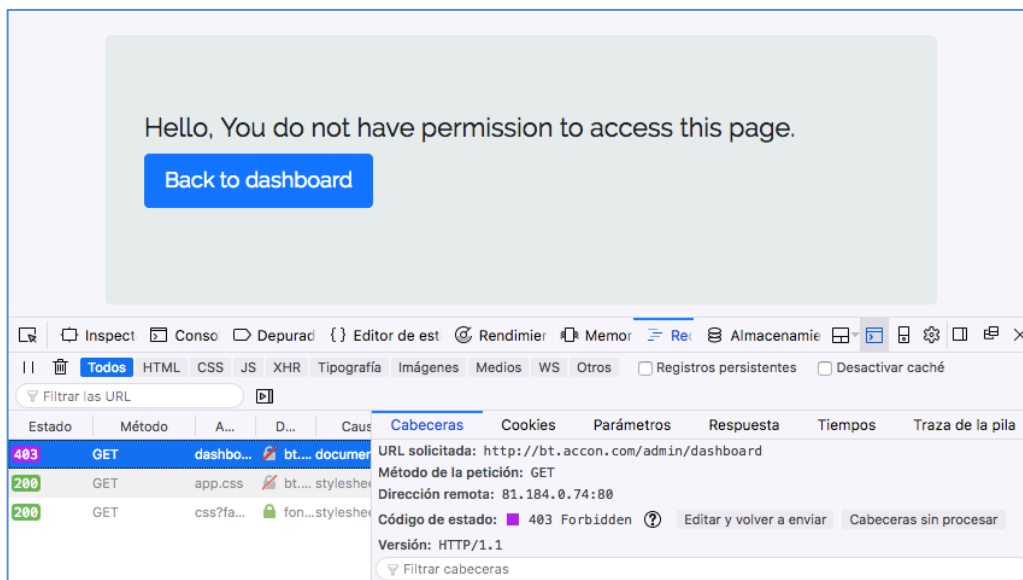


Figura 22: Ejemplo del middleware retornando el error 403

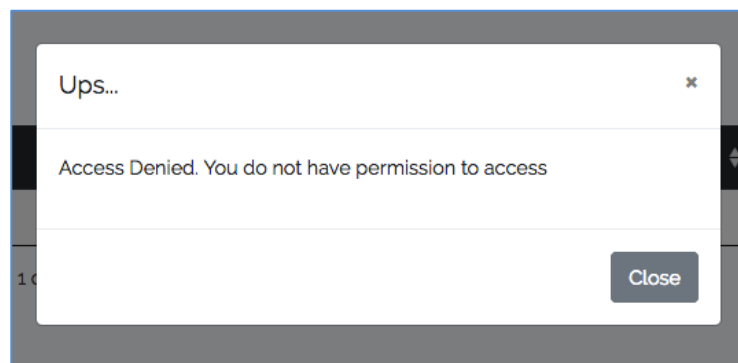


Figura 23: Ejemplo de middleware retornando un JSON

4.4 Usabilidad y diseño adaptativo

Se ha puesto especial énfasis en la usabilidad y accesibilidad de la aplicación web. El objetivo marcado es que el usuario final sea capaz de alcanzar sus objetivos con el mínimo esfuerzo, pero a la vez, obteniendo unos resultados óptimos.

La distribución del contenido se basa en una barra vertical, alojada a la parte izquierda de la pantalla, donde aparecerá todo el menú de la aplicación; y en una barra horizontal, en la parte superior de la pantalla, donde se muestran las notificaciones y la foto de perfil del usuario autenticado. En la parte central es donde mostraremos todo el contenido, de forma mayoritaria, en forma de lista o en paneles (widgets).

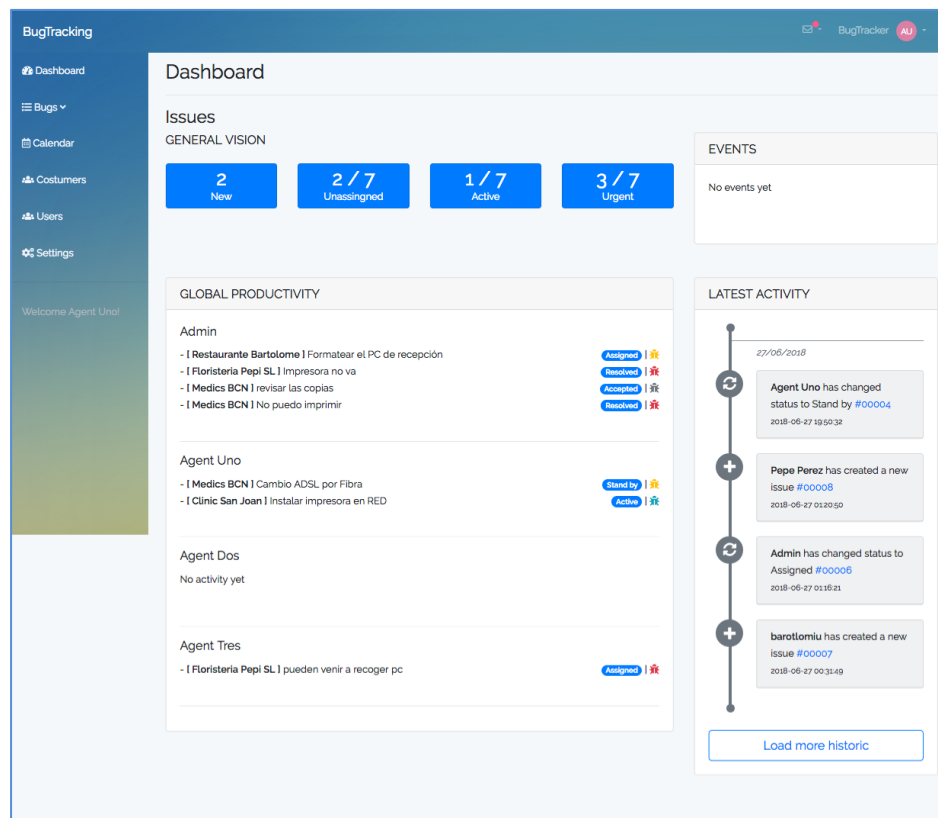


Figura 24: Página principal después de haberse autenticado en la aplicación

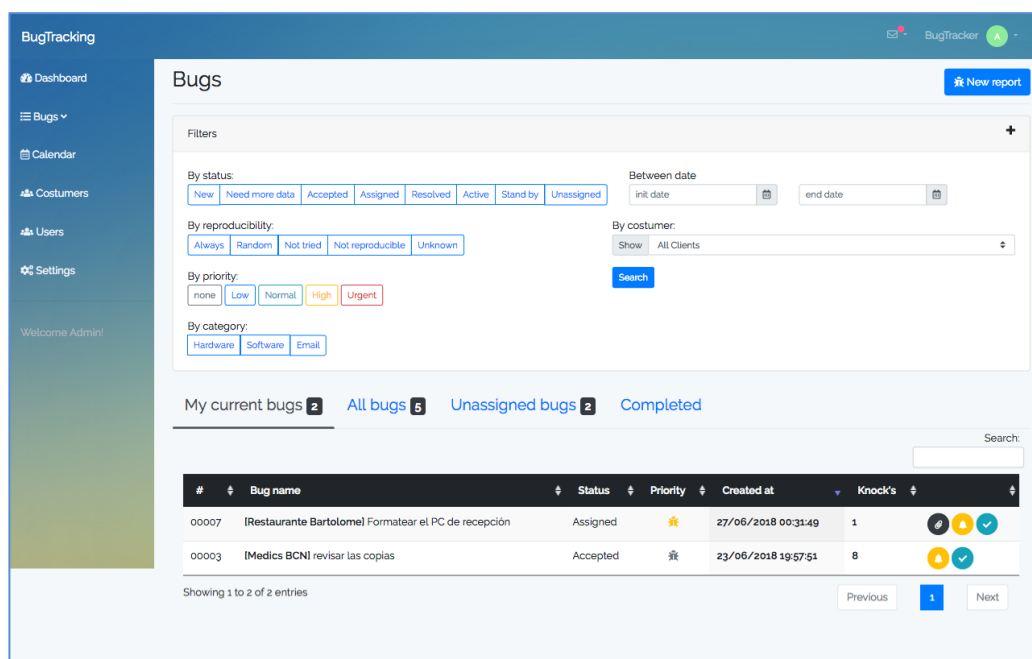


Figura 25: Página de incidencias

Por otra parte, se ha implementado en toda la aplicación el diseño adaptativo, debido a que la aplicación va dirigida a los agentes. Como se ha comentado anteriormente, hay veces que los agentes tienen que desplazarse a visitar a los clientes y, de esta manera, mientras se encuentran fuera de la oficina, tienen la posibilidad de conectarse desde su móvil o Tablet. El diseño adaptativo

permite el uso de la aplicación con un diseño adaptado a las dimensiones de la pantalla, disponiendo de las mismas opciones que disponemos accediendo desde el ordenador. En la *figura 26* se puede ver cómo sería la vista desde un dispositivo móvil:

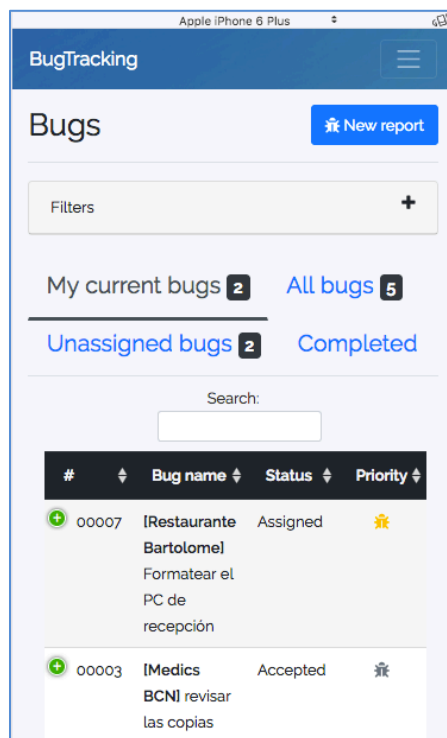


Figura 26: Página de Bugs vista desde un dispositivo móvil

Además, se puede observar que el menú principal ha quedado reducido y oculto en la parte superior del dispositivo debido a las dimensiones de la pantalla.

En la *figura 27* se puede observar cómo se adapta la vista a las dimensiones de una Tablet.

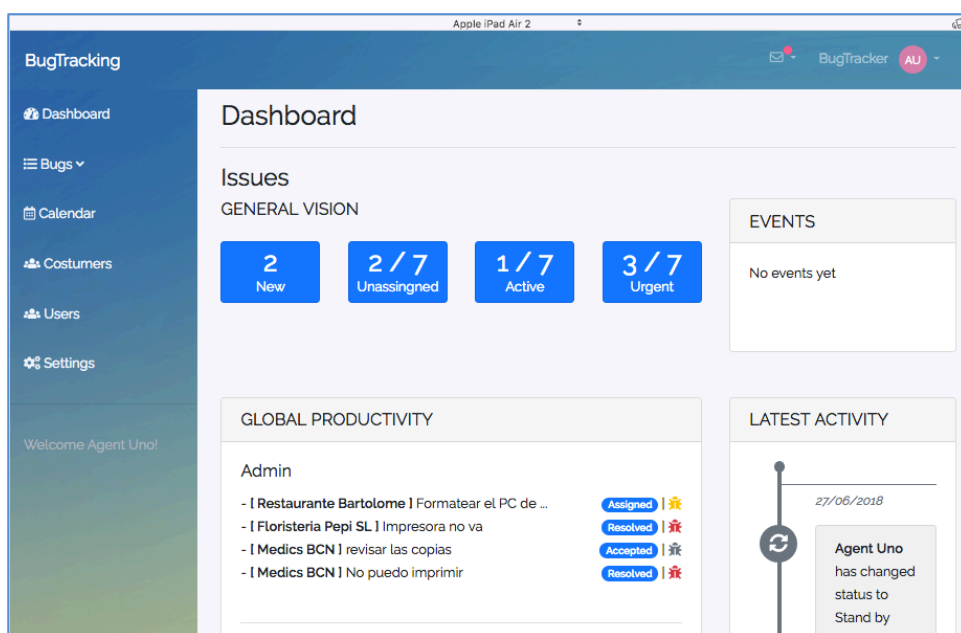


Figura 27: Página de dashboard vista desde una tablet

4.5 Sistema de CRUD

Las acciones: crear, leer, actualizar o borrar (de las siglas *Cread, Read, Update, Delete*) son básicas en esta aplicación. Con estas acciones podemos crear nuevos registros, modificarlos o borrar las incidencias, los usuarios, los eventos y los clientes en la aplicación.

Todas estas acciones se hacen a través de llamadas **Ajax**, que detallaremos en otro apartado.

A continuación, se mostrará el CRUD del módulo de incidencias:

Para **crear** una nueva incidencia debemos hacer clic en el botón izquierdo de la parte superior. Una vez hayamos hecho clic, se abrirá un pop-up con el formulario de crear una nueva incidencia (*figura 28*).

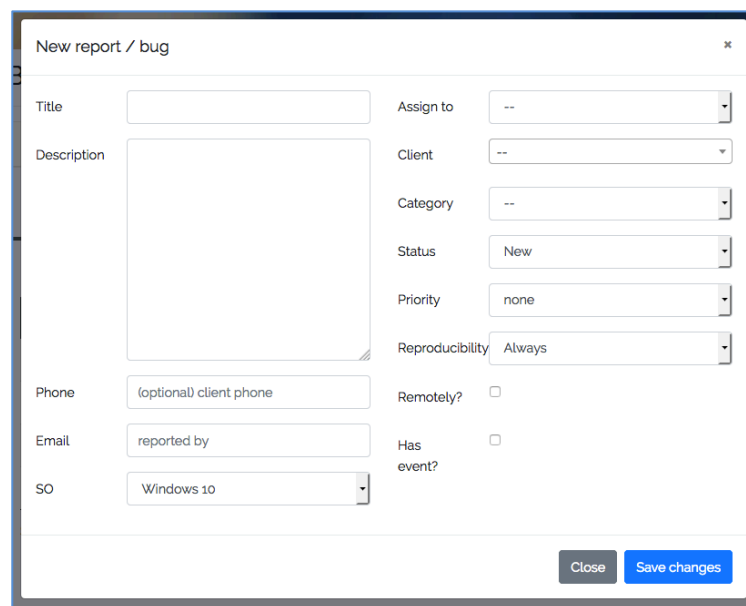











The image shows a web-based form titled "New report / bug" with a close button in the top right corner. The form is organized into two main columns. The left column contains a "Title" text input, a "Description" text area, a "Phone" input with a placeholder "(optional) client phone", an "Email" input with a placeholder "reported by", and an "SO" dropdown menu currently showing "Windows 10". The right column contains several dropdown menus: "Assign to" (showing "--"), "Client" (showing "--"), "Category" (showing "--"), "Status" (showing "New"), "Priority" (showing "none"), and "Reproducibility" (showing "Always"). Below these are two checkboxes: "Remotely?" and "Has event?", both of which are currently unchecked. At the bottom right of the form are two buttons: a grey "Close" button and a blue "Save changes" button.

Figura 28: pop-up para crear una nueva incidencia.

Para **leer** y/o **modificar** un registro de la aplicación se mostrará siempre un listado con todos los registros. Para acceder a uno de ellos, basta con hacer clic sobre el título de la incidencia. Una vez hecho el clic, se abrirá el pop-up con dicha información. Aparte de la información básica de la incidencia, se muestran las pestañas de información del cliente, actuaciones sobre la incidencia y ficheros adjuntos.

My current bugs 5 All bugs 5 Unassigned bugs 2 Completed						
Search:						
#	Bug name	Status	Priority	Created at	Knock's	
00017	[Floristeria Pepi SL] No recibe correos	Need more data	High	06/06/2018 22:30:05	1	  
00018	[Clinic San Joan] Formatear el PC de recepción	Active	Medium	06/06/2018 22:31:50	1	 
00019	[Medics BCNI] Instalar impresora en RED	Assigned	Medium	06/06/2018 22:33:01	1	 
00020	[Medics BCNI] Cambio ADSL por Fibra	Stand by	High	06/06/2018 22:34:00	1	 
00021	[Restaurante Bartolome] Montar camaras IP	Assigned	Medium	06/06/2018 22:34:53	1	 

Showing 1 to 5 of 5 entries

Previous **1** Next

Figura 29: Listado de incidencias

Report / bug

Bug info

Costumer info

Performances

Files attachment

Title

No recibe correos

Assign to

Agent Uno

Description

El cliente nos llama por qué no le estan llegando los correos.

Client

Floristeria Pepi SL

Category

Mail

Status

Need more data

Priority

High

Reproducibility

Always

Phone

63337373

Remotely?

☒

Email

pepe@floristeria.com

Has event?

☐

SO

Windows 7

Delete

Save changes

Figura 30: Pop-up con la información de una incidencia

Para **borrar** un registro, basta con abrir una incidencia y hacer clic en el botón *Delete*, seguidamente haremos la llamada al controlador para borrar las relaciones que tienen una incidencia tales como las actuaciones, el historial de estados, los ficheros adjuntos y después borraremos el registro.

Category

Delete bug?

You are sure you want to delete the incident, it will delete the attachments, notifications, history ... Do you want to delete the bug?

No

Yes

Figura 31: Pop-up de confirmación al borrar una incidencia

4.6 Subir ficheros

A la hora de añadir nuevas incidencias, a veces, el cliente necesita adjuntar una captura de pantalla del error en cuestión o alguna documentación extra que pueda servir de ayuda a los técnicos a la hora de resolverla. Por ello, se ha habilitado la posibilidad de subir ficheros y adjuntarlos a una incidencia, los formatos que permite la subida de ficheros son *.pdf*, *.jpeg*, *.jpg*, *.png*, *.doc*, *.docx*, *.xls* y *.xlsx*.

Estos ficheros se almacenan en el servidor en la carpeta */public/storage* y son accesibles a través de la carpeta */storage*.

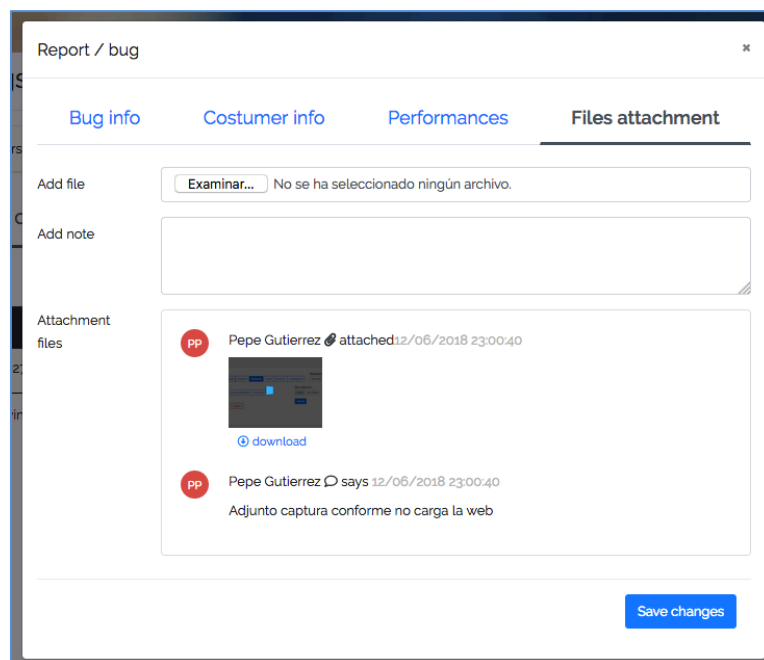


Figura 32: Listado de notas y ficheros adjuntos

Hay que destacar que, cuando hay algún fichero adjunto que haya subido el cliente desde su plataforma o el agente, estos tienen un pequeño icono de aviso tanto en el listado de todas las incidencias como a la hora de abrir una de ellas.

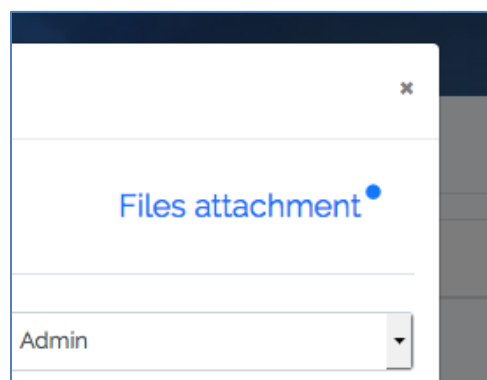


Figura 33: Icono de aviso de ficheros adjuntos al abrir una incidencia.

4.7 Llamadas asíncronas Ajax

Todo el contenido dinámico estará servido mediante la tecnología *AJAX*. Al ser una aplicación que va a estar en continuo movimiento es importante que las cargas de contenido no dificulten la experiencia del usuario.

Esta tecnología trabajará en el lado del cliente (navegador), por lo tanto, no ralentizaremos la web al conectarnos con el servidor.

Para realizar una búsqueda, tradicionalmente lo que se haría es pasar los parámetros por *GET*, es decir los parámetros enviados por la url, y el servidor nos devolvería los resultados refrescando toda la página web. Esto puede llegar a ser molesto si tenemos que ir actualizando la página cada vez que no encontremos resultados. Para evitar el ir refrescando la página y recargando recursos innecesariamente, con *AJAX* solo cargamos el contenido que estemos solicitando al servidor.

Pasamos los parámetros y hacemos la petición *GET* asíncronamente sin que el usuario final se dé cuenta y el resultado del servidor lo capturamos mediante el método *success(response)* de *AJAX*. Este método nos devuelve la respuesta en forma de *JSON* y mediante *jQuery* lo mostramos en *HTML*.

No obstante, si tenemos un gran registro de datos y no avisamos al usuario final que se está trabajando en ello, por ejemplo a la hora de buscar, haremos uso del método *beforeSend()* para avisar que se está realizando su petición.

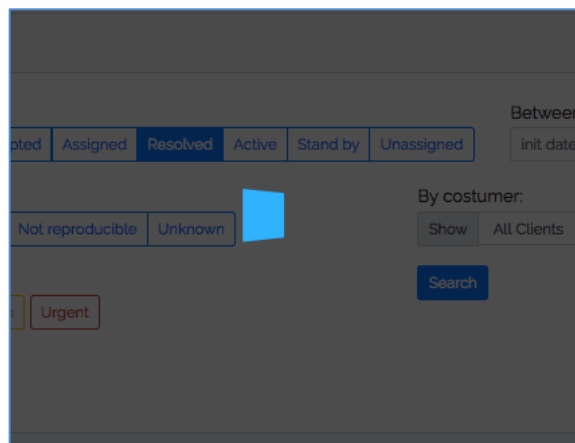


Figura 34: Loader spinner, elemento CSS

Para las acciones de enviar datos y que el servidor los guarde también haremos uso de *AJAX*. Cuando abrimos el *popup* de nuevo registro enviamos nuestros datos por *POST*; si se diera algún error al pasar los datos, en este caso, el servidor nos devolverá un mensaje y lo mostraremos sin necesidad de volver a recargar todos los recursos (Figura 35).

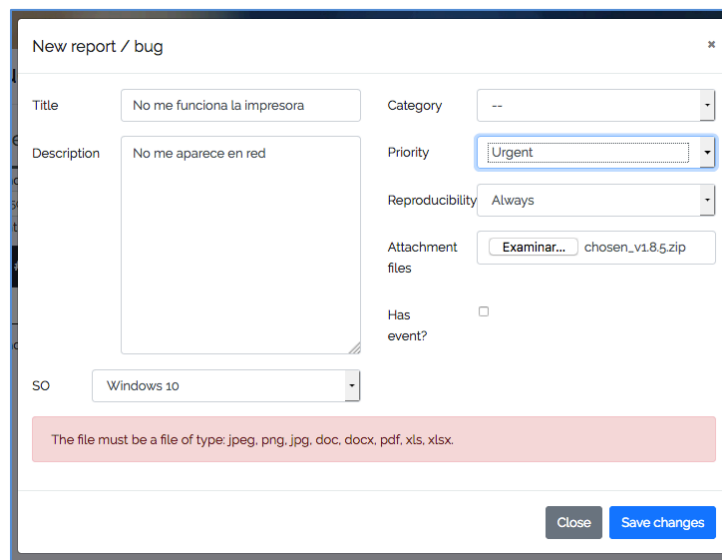


Figura 35: Respuesta del servidor validando la entrada de datos

4.8 Seguridad

En todos los formularios donde el usuario final entra datos al sistema, se ha prevenido la seguridad contra inyecciones *SQL*, que pueden poner en peligro los datos de la aplicación, y los ataques de tipo *XSS*, que habíamos comentado anteriormente, que pueden alterar la integridad de nuestra aplicación.

Veamos un ejemplo de *XSS*: en este caso, hemos entrado como título un script con un *alert* de JavaScript, pero, también, podemos hacer una redirección a otra web, cambiar el contenido de la web o la incrustación de *iframes*.

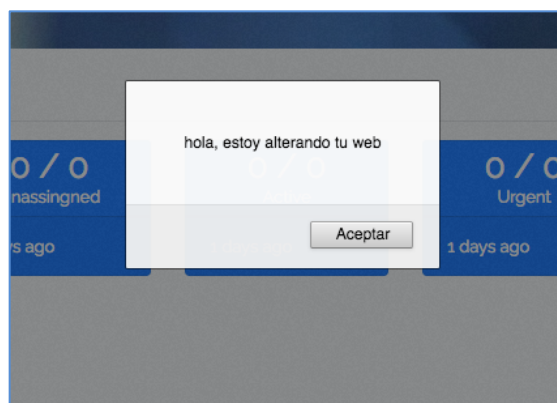


Figura 36: Ejemplo de ataque *XSS* de JavaScript

4.9 Permisos

Los permisos de acciones o accesos a un recurso de la aplicación pasarán primero por una pequeña comparación de permisos como los de administrador, agente, informador y cliente que se han definido en la aplicación.

Cuando un usuario determinado inicia sesión, dependiendo del perfil, se mostrarán diferentes opciones. En el caso de un administrador, por ejemplo, tendrá acceso total a todas las funcionalidades, en cambio, en el caso de un cliente su acceso estará limitado.

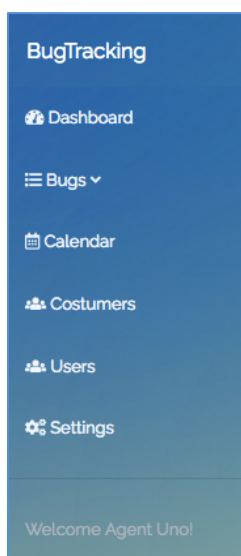


Figura 37: Menú de navegación
del perfil administrador o agente

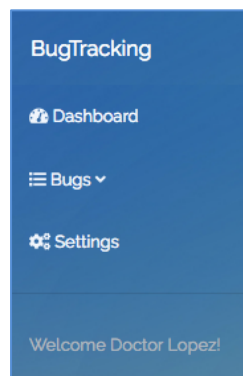


Figura 38: Menú de navegación
del perfil cliente

No obstante, estos son simples condiciones en la vista, para que en la página no se muestren dichas opciones, pero aparte de esto, también lo controlamos a nivel de middleware donde se filtran las peticiones y se decide si dar o no acceso.

4.10 Eventos / Notificaciones

Los eventos son muy útiles en la aplicación. Estos se han implementado para poder escuchar todo lo que pasa dentro de la misma como, por ejemplo: cuando una incidencia la ha creado un cliente, cuando asignamos una incidencia a un agente o cuando creamos un evento en el calendario.

Esto es muy útil ya que, de alguna manera, nos permite separar la lógica de la aplicación y crear un evento con varios "oyentes" por separado y mantener limpio nuestro código.

Por ejemplo, cuando una incidencia se crea en la aplicación queremos hacer varias gestiones en ella como, por ejemplo, añadirlo al histórico y notificar (tanto por la aplicación como por e-mail). Por ello, llamamos al evento **BugCreated**, que tiene varios “oyentes” que se encargarán de escuchar y aplicar esta lógica.

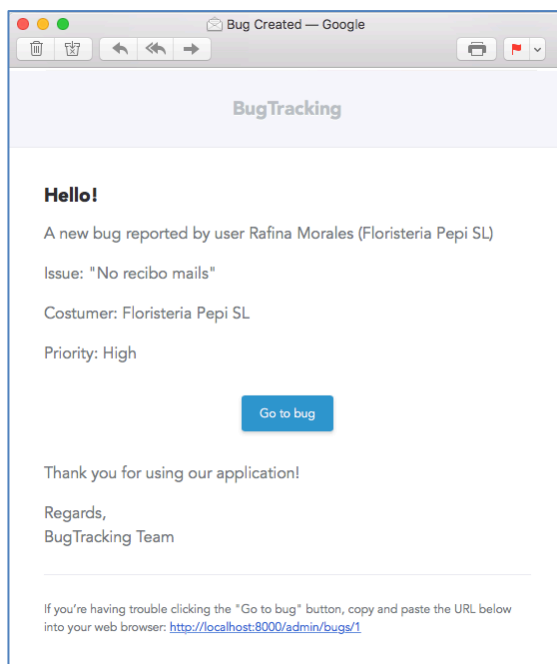


Figura 39: E-mail de notificación cuando una incidencia se ha creado

Notificaciones

Respecto a las notificaciones de la aplicación, cuando el agente esté autenticado y tenga algún tipo de notificación, la aplicación actualizará el título del navegador con el número de notificaciones que tenga y avisará con un breve sonido.

También, en la barra de navegación, en el icono del “sobre” aparecerá un símbolo que avisa de las notificaciones pendientes de ver.

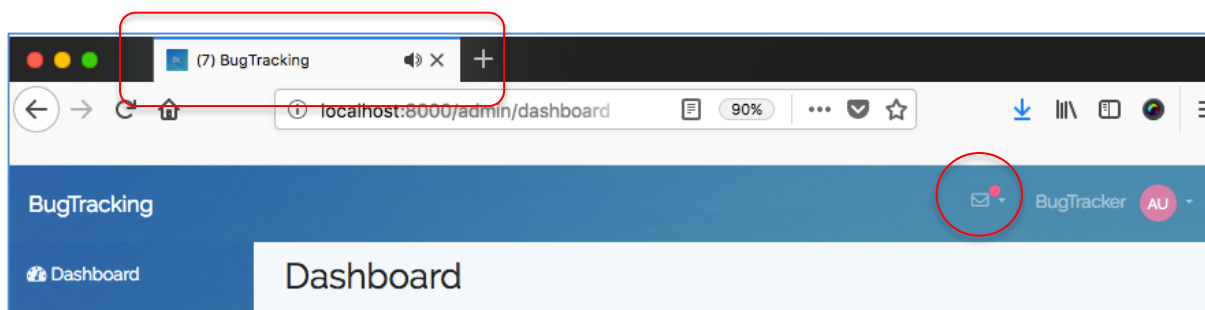


Figura 40: Aviso de notificaciones

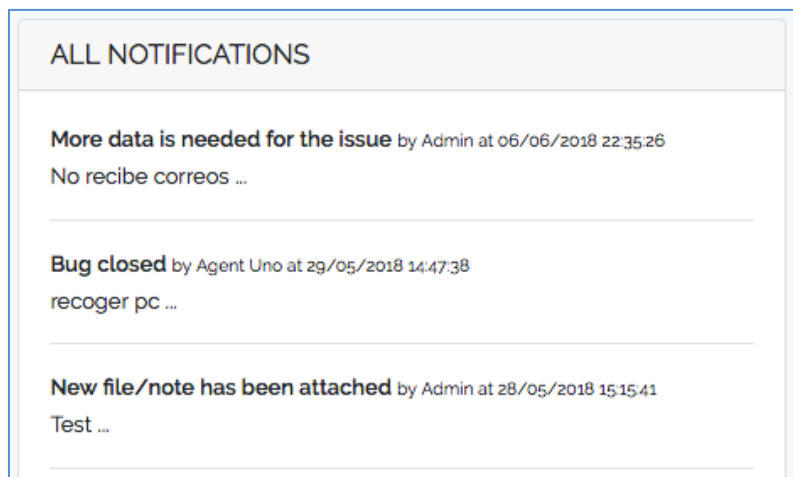


Figura 41: Notificaciones de un cliente en la aplicación

4.11 Componentes de la aplicación

Listados

En lo que se refiere la gestión de incidencias y/u otros apartados donde se necesite la visualización de todos los registros seguirán el mismo formato. Todas ellas se mostrarán en forma de lista con diferentes opciones que el usuario podrá realizar.

En este caso, en las gestiones de incidencias, se muestra un listado con la información básica de las incidencias como el nombre, el estado, la prioridad, cuando se ha creado, las veces que el cliente nos ha dado un toque y las opciones rápidas que se pueden realizar sin la necesidad de abrir una incidencia (por ejemplo, acceder rápidamente a los archivos adjuntos, añadir un nuevo toque o completar la incidencia).

My current bugs 5 All bugs 5 Unassigned bugs 2 Completed							
Search:							
#	Bug name	Status	Priority	Created at	Knock's		
00017	[Floristeria Pepi SL] No recibe correos	Need more data	🔴	06/06/2018 22:30:05	1	🔔	✅
00018	[Clinic San Joan] Formatear el PC de recepción	Active	🟢	06/06/2018 22:31:50	1	🔔	✅
00019	[Medics BCN] Instalar impresora en RED	Assigned	🟡	06/06/2018 22:33:01	1	🔔	✅
00020	[Medics BCN] Cambio ADSL por Fibra	Stand by	🔴	06/06/2018 22:34:00	1	🔔	✅
00021	[Restaurante Bartolome] Montar camaras IP	Assigned	🟡	06/06/2018 22:34:53	1	🔔	✅
Showing 1 to 5 of 5 entries							
				Previous	1	Next	

Figura 42: Listado de registros de las incidencias

Buscadores

Los listados disponen de un buscador sencillo, se trata de un campo de texto justo arriba de cada tabla. Este buscador, usa la tecnología de **jQuery** que hace la búsqueda en el lado del cliente sin hacer ninguna llamada al servidor, por lo que la manera de buscar es rápida.

Además, dispone de una búsqueda avanzada con la finalidad de simplificar el tiempo a la hora de realizar la búsqueda de incidencias.

Este sí que hace uso a llamadas del servidor ya que podemos filtrar por estados, por reproducibilidad, por prioridades, categorías, por un cliente en específico y sobre todo por fecha o periodos de fechas.

Aún que como hemos comentado, las llamadas son al servidor, pero al hacer uso de Ajax las llamadas no dificultan a la usabilidad del usuario.

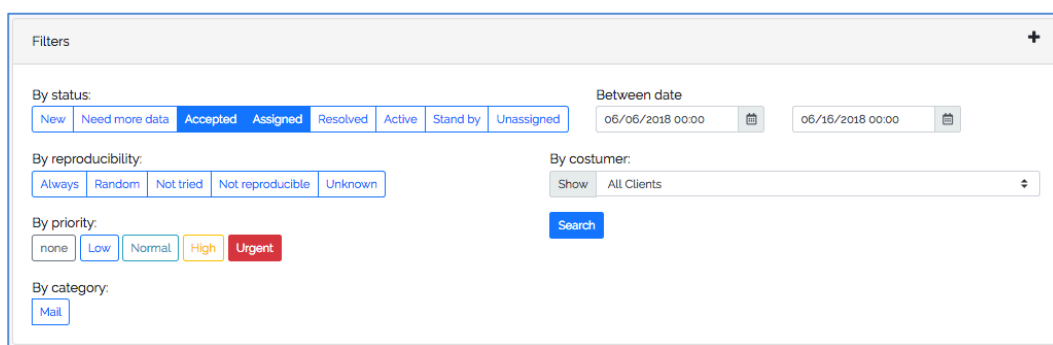
The image shows a 'Filters' panel with a '+' icon in the top right corner. It contains several filter sections: 'By status' with buttons for New, Need more data, Accepted, Assigned, Resolved, Active, Stand by, and Unassigned; 'Between date' with two date pickers showing 06/06/2018 00:00 and 06/16/2018 00:00; 'By reproducibility' with buttons for Always, Random, Not tried, Not reproducible, and Unknown; 'By priority' with buttons for none, Low, Normal, High, and Urgent; 'By category' with a button for Mail; and 'By customer' with a dropdown menu showing 'Show' and 'All Clients'. A 'Search' button is located at the bottom right of the filter panel.

Figura 43: Buscador avanzado de incidencias

Widgets

Los widgets se usarán para mostrar información rápida al usuario. Su utilidad se basa en mostrar una pincelada o datos importantes de información sobre eventos próximos, actividades que hace el usuario, etc.

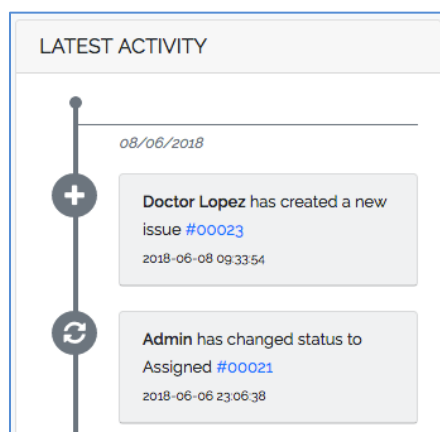


Figura 44: Widget timeline

Calendario

Al trabajar con fechas en la aplicación sobre los requerimientos y movimientos que hacen los agentes, se ha implementado un calendario común para todos los trabajadores donde su principal tarea es mostrar, en forma de calendario, los eventos programados de cada uno de ellos. Aquí el usuario podrá añadir nuevos eventos del tipo reunión, vacaciones u otros. También, se mostrarán aquí las incidencias programadas con una fecha en concreto.

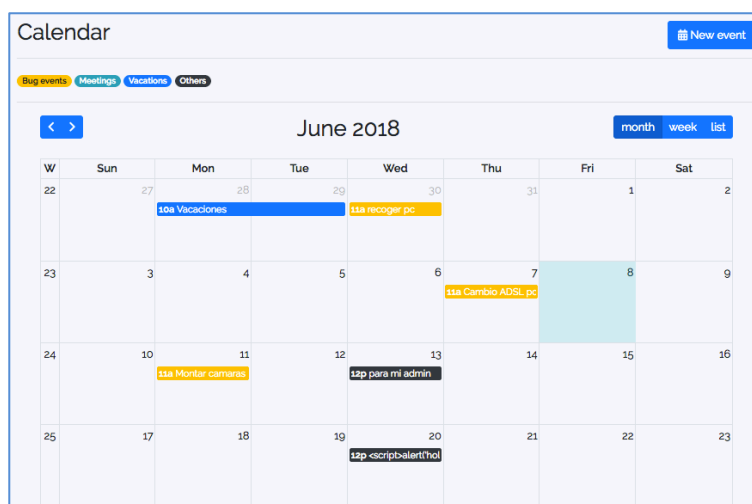


Figura 45: Pantalla general del calendario

Pop-ups

La manera tradicional de cualquier aplicación para mostrar la información detallada de un registro es redirigir al usuario a una nueva vista con toda la información y, después, se tiene que volver a la vista anterior para consultar todo el listado de registro y/o acceder a otro registro.

Para solucionar este tipo de navegabilidad y hacerlo más cómodo y rápido al usuario se han implementado las vistas parciales o pop-ups. Esta solución, evita todo esto haciendo que la información se abra en la misma página en forma de diálogo o **Modal**.

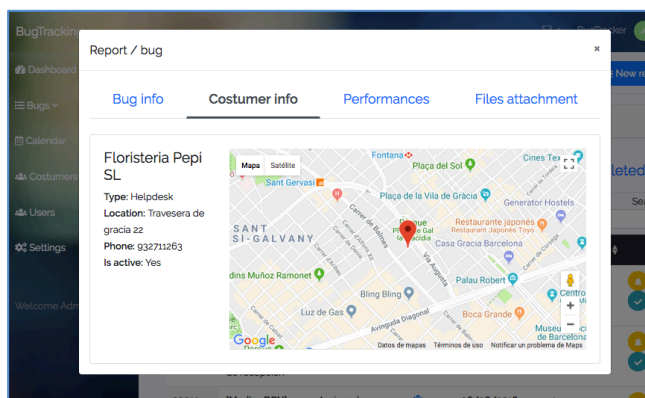


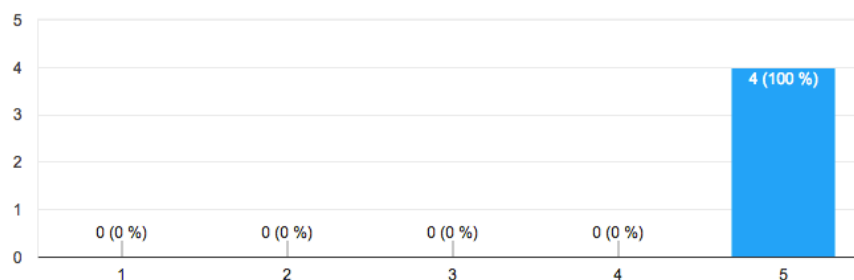
Figura 46: Modal mostrando la información del cliente

5. Pruebas y resultados

Después de la fase de implementación se llevó a cabo la puesta en marcha de la aplicación a una empresa IT que ofrece el servicio SAT. Tras dos semanas de pruebas en un entorno real, 4 técnicos pudieron probar con total libertad la aplicación y posteriormente se ha realizado un cuestionario para saber el grado de satisfacción de los objetivos establecidos.

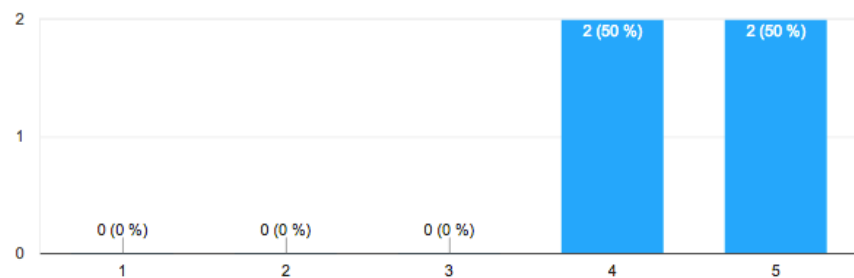
1. La aplicación ha cumplido con mis expectativas en el momento de gestionar una incidencia.

4 respuestas



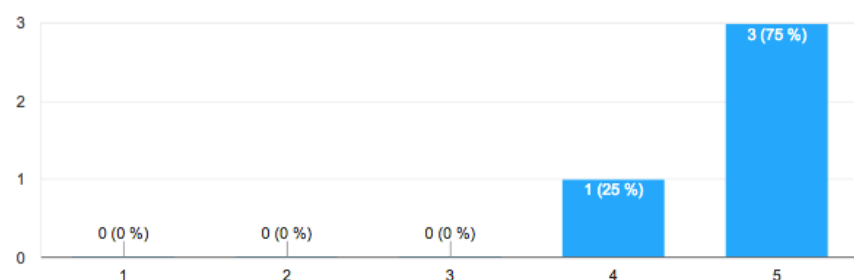
2. Considero que la aplicación ha permitido que la productividad de los trabajadores mejore.

4 respuestas



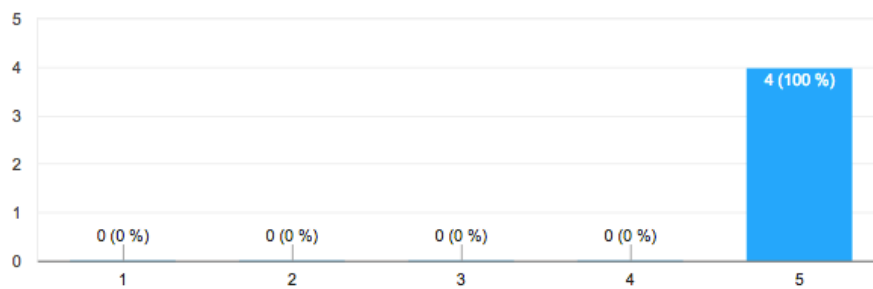
3. En cuanto a su uso, la experiencia de usuario te ha sido satisfactoria.

4 respuestas



4. La aplicación me ha ayudado a organizar mi trabajo y/o el de mis compañeros.

4 respuestas



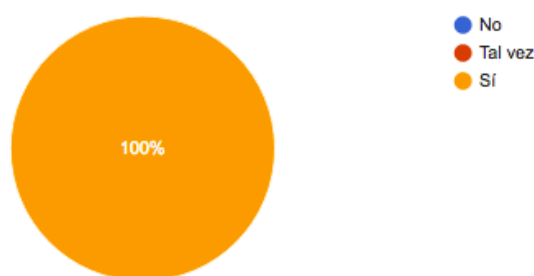
5. ¿Crees que es interesante que el cliente pueda estar al tanto del estado de su incidencia?

4 respuestas



6. ¿Recomendarías esta aplicación?

4 respuestas



Como se puede observar los 4 técnicos están de acuerdo de que la gestión de incidencias ha cumplido con sus expectativas y ha ayudado a la organización de sus tareas y/o la de sus compañeros.

Por otro lado, el 50% de los técnicos creen que en ocasiones es conveniente de que el cliente esté al tanto del transcurso de su incidencia y el otro 50% piensa que puede ser beneficioso para ambas partes.

Y, por último referente a la usabilidad de la aplicación, el 75% de los técnicos creen que sí posee la aplicación con una buena usabilidad y el 25% restante cree que debería aún mejorarse.

6. Conclusiones

Considero que los objetivos específicos descritos en la introducción de este Proyecto Final de Grado han sido satisfactoriamente cumplidos gracias a los conocimientos previos adquiridos en el grado, así como la experiencia en el desarrollo de software que he conseguido en el transcurso del desarrollo de la aplicación.

He podido profundizar de manera positiva todo lo relacionado con la arquitectura de software y, sobre todo, la importancia que tiene un buen diseño en la arquitectura, ya que ayuda a que la implementación sea más fácil. Sobre el uso de un *framework* y el conjunto de tecnologías que lo envuelven, he podido comprobar que usar el patrón MVC hace que el programa sea modutable y escalable para futuras ampliaciones o mejoras.

En lo que respecta con la aplicación web y los objetivos generales, se ha conseguido agilizar el tiempo de respuesta ya que la mayor parte de gestión la hacemos en el lado del navegador y sólo recurrimos al servidor para consumir datos o insertar nuevos registros. También, la usabilidad de la aplicación, tanto en escritorio como en dispositivos móviles, ha ayudado a que la usabilidad de usuario sea cómoda y fácil de usar.

Por otro lado, se ha conseguido agilizar aún más las tareas de los técnicos gracias al acceso de los clientes en la plataforma para que ellos mismo puedan crear incidencias y, de esta manera, los técnicos ahorran el tiempo de crearlas.

Otro punto favorable de la aplicación es el hecho que sea multiplataforma, es decir, la podrán usar fuera de la oficina con cualquier dispositivo que tenga conexión a internet. Esta característica es importante ya que los técnicos a veces están en movimiento y necesitan consultar información.

La productividad de trabajo es mucho mejor ya que se puede reasignar incidencias para que otro técnico la continúe y gracias a las actuaciones o notas que pueden adjuntar los clientes, el técnico reasignado sabrá por donde continuar.

Finalmente, quiero resaltar la puesta en marcha de este proyecto en un entorno real dónde técnicos informáticos han hecho uso de la aplicación y se ha podido comprobar que la herramienta cumple con todos los requisitos iniciales.

6.1 Futuras ampliaciones

Gracias a la puesta en marcha se han detectado mejoras para poder integrar al 100% esta aplicación en un entorno real y funcional.

En lo que respecta a la gestión de incidencias sería interesante añadir geolocalización del dispositivo. Esta ampliación sería muy útil ya que, si la incidencia requiere el desplazamiento, el técnico podría consultar su ubicación en el mapa y podría generar una ruta hacia la localización del cliente.

Por otro lado, se podría implementar la facturación de las incidencias una vez finalizadas ya que el sistema guarda todas las actuaciones que se han llevado a cabo y notifica vía email al cliente del cierre de la incidencia. Sería interesante que se pudiera emitir automáticamente la factura y se cargara el recibo por el banco de forma automática.

Por último, en cuanto a la aplicación web, sería interesante implementar un sistema de notificaciones *Push*. Los técnicos son notificados vía email sobre alguna actualización o nuevo registro de incidencias, pero si el agente no está pendiente del email o no está autenticado en la aplicación no podrá saber si hay nuevas incidencias. Por ello, con las notificaciones *Push* podríamos avisarles y así estarían pendientes sobre lo que ocurre en cada momento.

Para finalizar, es importante tener seguridad con las transferencias de datos que viajan entre el navegador y el servidor. Actualmente, estamos usando el protocolo *HTTP*, esto quiere decir que las transferencias no tienen ningún tipo de seguridad y podríamos sufrir ataques del tipo “*man in the middle*” donde podemos correr el riesgo de que el atacante obtenga credenciales o información confidencial. Debido a esto, se podría implementar el protocolo *HTTPS*, de tal manera que los datos se transferirían cifrados y podríamos prevenir estos ataques.

7. Glosario

AJAX: *Asynchronous JavaScript And XML*. Es una técnica de desarrollo de web para crear aplicaciones interactivas sin la necesidad de volver a cargar el contenido.

Artisan: Es una interfaz de línea de comandos que ofrece un conjunto de comandos que ayudan a realizar diferentes tareas durante el desarrollo de la aplicación

Backoffice: En el entorno IT hace referencia a todas las tareas de la gestión de una empresa.

Framework: Es un patrón o esqueleto para el desarrollo y/o implementación de una aplicación.

CSS: *Cascading Style Sheets*. Es un lenguaje de diseño que permite tener una apariencia agradable a la hora de diseñar una página web.

HTML: *HyperText Markup Language*. Lenguaje de marcas para la elaboración de páginas web.

Javascript: Es un lenguaje de programación interpretado en el lado del cliente, permite mejorar la interfaz de usuario y hacer páginas dinámicas.

JSON: *Javascript Object Notation*. Es un formato de texto para el intercambio de datos.

Jquery: Es una biblioteca de Javascript, que su primera función es simplificar la manipulación DOM, de AJAX y manejo de eventos.

MVC: *Modelo, vista, controlador*. Es un estilo de arquitectura de software que separa la lógica de negocio, la interfaz del usuario y la persistencia de datos.

MYSQL: Sistema de gestión de base de datos relacional.

PHP: *Hypertext preprocessor*. Es un lenguaje de programación de código abierto, utilizando para el desarrollo de aplicaciones web. Interpretado a lado del servidor, pero se puede incrustar en HTML.

Responsive web design: Diseño de web adaptable, es una filosofía de diseño web cuyo objetivo es adaptar las vistas HTML a los diferentes tamaños de los navegadores y móviles.

SAT: *Servicio de Asistencia Técnica*. Hace referencia al departamento o empresa que ofrece un servicio de atención técnica, en este caso para el sector IT.

Template: Es un conjunto de archivos que determinan la estructura y el aspecto de una aplicación web

ORM: *Object Relational Mapping*. Es una técnica de programación para convertir datos entre el lenguaje de programación y la utilización de una base de datos.

Widget: En el entorno IT, es una pequeña aplicación donde se muestra contenido en algún lugar de la página web.

8. Bibliografía

Bootstrap. (s.f). *Documentation v4.1*. Recuperado el 1 de junio del 2018 de

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

Joanna Ngai (2017). *How to be a Design Usability Champ*. Recuperado el 8 de marzo del 2018 de

<https://webdesign.tutsplus.com/articles/how-to-be-a-design-usability-champ--cms-29610>

Kscarfone & more. (2015). MVC architecture. Recuperado el 15 de marzo del 2018 de

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)

[US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)

Mehdi Achour & more. (2018). *PHP Manual*. Recuperado 2 de abril del 2018 de

<http://php.net/manual/en/index.php>

Laravel. (s.f.). *Documentación oficial Laravel 5.6*. Recuperado 22 de marzo del 2018 de

<https://laravel.com/docs/5.6>

The jQuery Foundation. (s.f). *Ajax*. Recuperado 5 de abril del 2018 de

<http://api.jquery.com/category/ajax/>

Anexo. Manual de usuario

Este manual va dirigido a los trabajadores que prestan el servicio SAT y para los clientes que usarán esta herramienta como plataforma para registrar incidencias.

Manual para usuarios SAT

1. Acceder a la aplicación

Para hacer uso de la aplicación accede a la siguiente URL: <http://bt.accon.com/>

Seguidamente, introduce tu usuario y contraseña:

Usuario: agent@bugtracking.com

Contraseña: 12345aA

Una vez introducido los datos de acceso, haz clic en el botón *Login*.

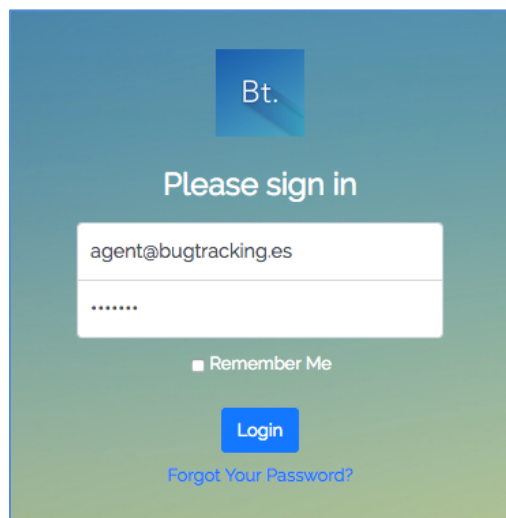


Figura 21: Formulario de acceso a la plataforma

2. Crear una Incidencias

En este apartado se mostrará como registrar una nueva incidencia.

Para ello, en el menú lateral de la parte izquierda, haz clic en *Bugs*, y seguidamente en *Bug List*.

Una vez dentro, haz clic en el botón *New report*.

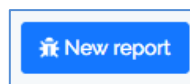


Figura 49: Botón para registrar una incidencia

A continuación, se mostrará el siguiente pop-up dónde introduciremos toda la información necesaria.

Los campos requeridos son:

1. Título: describir el título de la incidencia
2. Descripción: describir el problema de la incidencia.
3. Teléfono: de la persona que reporta la incidencia
4. Email: de la persona que reporta la incidencia
5. SO: sistema operativo
6. Asignar a
7. Cliente
8. Categoría
9. Estado
10. Prioridad
11. Reproducibilidad
12. Es en remoto
13. Es evento
 - a. Fecha inicio
 - b. Fecha fin

El resto de los campos no son obligatorios, pero se recomienda rellenarlos para así tener la máxima información de la incidencia.

Para finalizar, haz clic en *Save changes*.

The screenshot shows a web form titled "New report / bug". The form is divided into two main sections. The left section contains a "Title" text input field and a "Description" text area. The right section contains several dropdown menus: "Assign to", "Client", "Category", "Status" (set to "New"), "Priority" (set to "none"), and "Reproducibility" (set to "Always"). Below these are checkboxes for "Remotely?" and "Has event?". At the bottom left, there are input fields for "Phone" (with a placeholder "optional client phone"), "Email" (with a placeholder "reported by"), and a dropdown for "SO" (set to "Windows 10"). At the bottom right, there are two buttons: "Close" and "Save changes". A red rectangle highlights the "Title" and "Description" fields, and a red arrow points to the "Save changes" button.

Figura 50: Formulario para crear una nueva incidencia

¿Cómo cambiar el estado de una incidencia?

Para ello, haz clic sobre una incidencia disponible en el listado de *My current bugs* o *All bugs*.

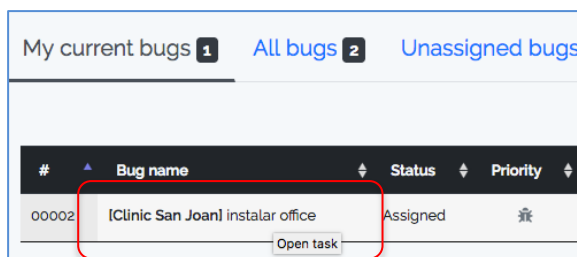


Figura 51: Listado de incidencia con el tooltip

Seguidamente se abrirá la incidencia en forma de pop-up.

Una vez abierto, en el selector de *Status*, selecciona el estado y para finalizar haz clic en el botón *Save changes*.

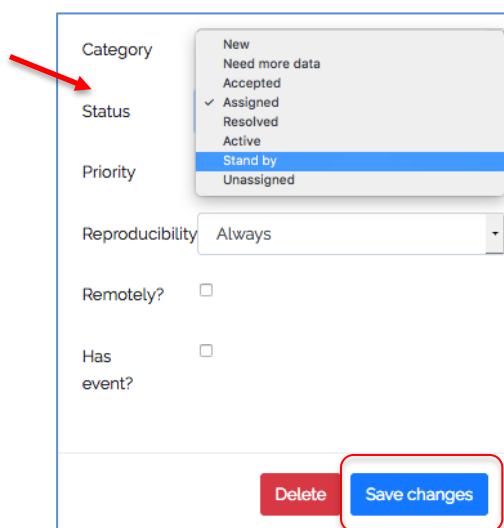


Figura 52: Se muestra el selector para cambiar el estado de la incidencia

Asignar o re-asignar incidencia

Para asignar o reasignar una incidencia, existen varias formas de hacerlo.

Lo podemos hacer cuando se está registrado una nueva incidencia, si queremos modificarla, consultarla.

Para ello, en el pop-up de la incidencia, seleccionamos a quien asignamos la incidencia en el selector de *Assign to*.

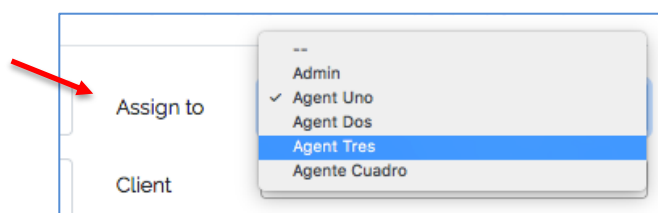


Figura 53: selector para cambiar de asignación

Desde el apartado de administración

También existe la posibilidad de reasignar las incidencias. Para ello, en el menú lateral haz clic en *Bugs* y seguidamente en *Bug settings*.

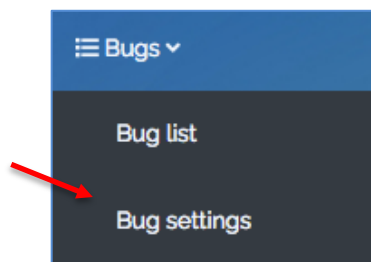


Figura 54: Submenú de Bugs

A continuación, en el widget *Reassing bugs*, se muestra dos columnas. Una de ellas es los usuarios con sus incidencias y en la otra columna los técnicos a los que reasignar la incidencia. Para hacer la reasignación, sigue los siguientes pasos:

1. En la columna izquierda, haz clic sobre un trabajador, seguidamente se muestra sus incidencias. Haz clic en las incidencias a reasignar.
2. Seguidamente, en la columna derecha, selecciona al trabajador.
3. Y por último, haz clic en el botón *Reassign*.

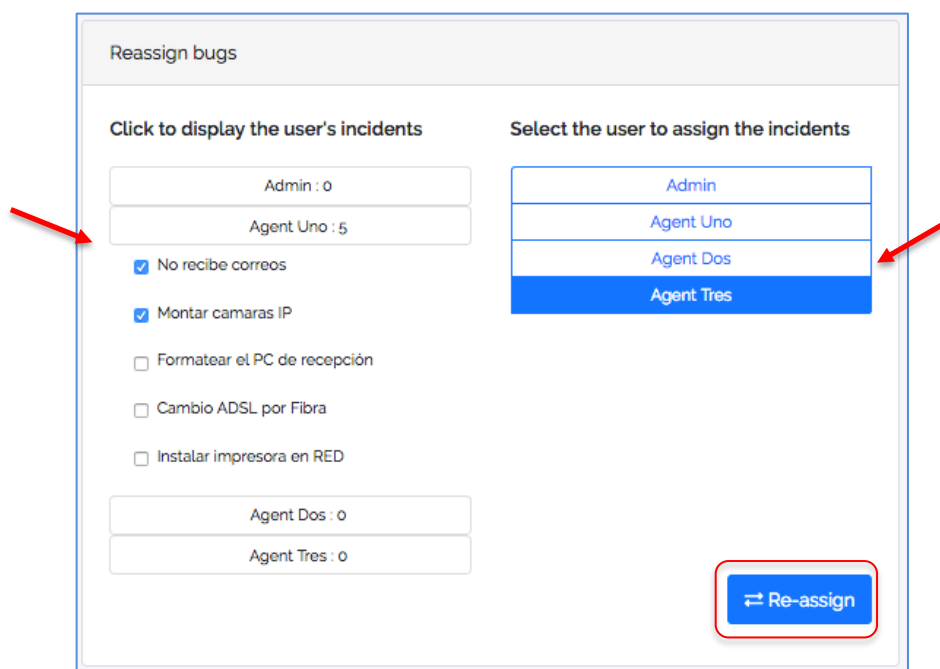


Figura 55: Volcado de re-asignación de incidencias

Cerrar una incidencia

En este apartado, veremos cómo dar por finalizada una incidencia.

Se puede hacer de dos formas, una de ellas es abrir la incidencia y cambiar el estado, como se ha comentado en el apartado ¿Cómo cambiar el estado de una incidencia?

La otra manera es desde el listado de incidencias. Para ello, en la última columna, haz clic en el botón *Check*.

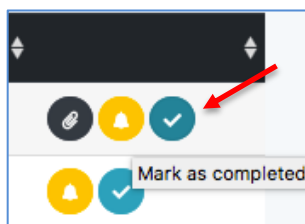


Figura 56: Opciones de una incidencia.

3. Gestionar calendario

En este apartado, podemos consultar los eventos programados de los técnicos y también podemos registrar nuevos eventos.

Para acceder al apartado de Calendario, haz clic en el menú lateral en la opción *Calendar*.

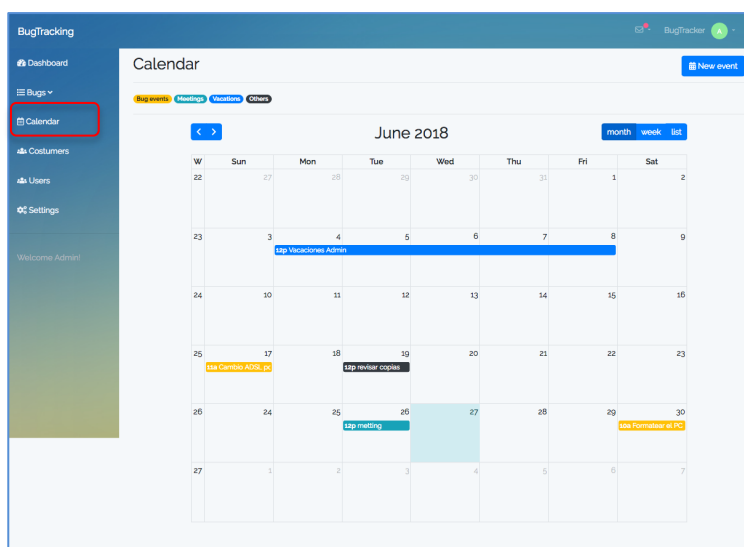


Figura 57: Pantalla de calendario

Registrar un evento

Para ello, haz clic en el botón *New event*, que se encuentra en la parte superior de la pantalla. Seguidamente se mostrará un pop-up con los siguientes campos a rellenar:

1. Título: título del evento
2. Descripción: descripción del evento
3. Asignar a: a que persona queremos asignar este evento
4. Tipo: tipo del evento, puede ser reunión, vacaciones u otros.
5. Empieza: fecha y hora del evento
6. Finaliza: fecha y hora del evento

Una vez rellenados, haz clic en el botón *Save changes* para registrar el evento.

Figura 58: Pop-up para registrar un nuevo evento

4. Gestionar clientes

En este apartado podemos ver todos los clientes registrados en la plataforma y crear un nuevo cliente.

Para acceder al apartado de Clientes, haz clic en el menú lateral en la opción de *Costumers*.

#	Name	Direction	Phone	Is Active	
1	BugTracker	Calle falsa 123	93273373	Yes	Edit
2	Floristeria Pepi SL	Travesera de gracia 22	932711263	Yes	Edit Delete
3	Medics BCN	Carrer aribau 23	937445993	Yes	Edit Delete
4	Restaurante Bartolome	Carrer ballen 2	937761120	Yes	Edit Delete
5	Clinic San Joan	Carrer muntaner 523	943223889	Yes	Edit Delete

Figura 60: Pantalla de clientes

Registrar cliente

Para ello, haz clic en el botón *New costumer*. Que se encuentra en la parte superior de la pantalla.

Seguidamente se mostrará un pop-up con los siguientes datos a rellenar:

1. Nombre: nombre del cliente
2. Descripción: información relevante del cliente
3. Teléfono
4. Tipo de suscripción: los diferentes tipos de servicios
5. Localización: para añadir la geolocalización del cliente

6. Si es activo

Para guardar al cliente haz clic en el botón *Save changes*.

The 'Costumer' form includes the following fields: Name (Client name), Description (Client description), Phone number (Client phone), Type suscription (dropdown menu), Localización (Google Map of Barcelona area), Location (Location client), and Is active? (checkbox). At the bottom right, there are 'Close' and 'Save changes' buttons, with the latter highlighted by a red rectangle.

Figura 61: Pop-up para registrar un nuevo cliente

5. Gestionar usuarios

En este apartado podemos ver todos los usuarios registrados en la plataforma y crear un nuevo usuario.

Para acceder al apartado de Usuarios, haz clic en el menú lateral en la opción *Users*.

The 'User list' page shows a table of users with columns: #, Name, Email, Client, and Phone. There are three users listed. A 'New user' button is in the top right. The 'Users' option in the left sidebar is highlighted with a red rectangle.

#	Name	Email	Client	Phone
5	Pepe Perez	pepe@floristeria.com	Floristeria Pepi SL	
6	Doctor Lopez	lopez@medics.com	Medics BCN	--
7	barotiomliu	pepelots@bar-tolome.com	Restaurante Bartolome	98374675647

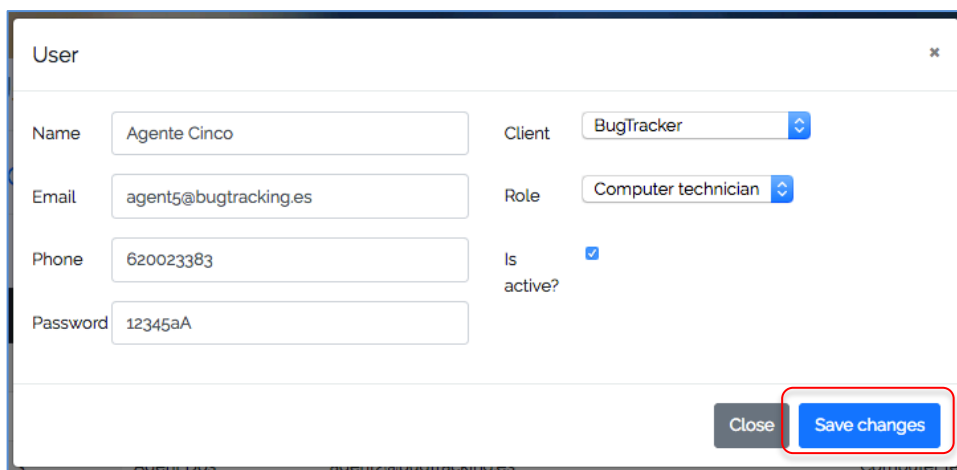
Figura 62: pantalla de usuarios

Registrar usuario

En este apartado podemos registrar usuarios con diferentes perfiles dentro de la aplicación. Para ello, haz clic en el botón *New user*. Que se encuentra en la parte superior de la pantalla. Seguidamente se mostrará un pop-up con los siguientes datos a rellenar:

1. Nombre
2. Email
3. Teléfono

4. Contraseña
5. Cliente: seleccionar el cliente
6. Role: seleccionar el rol en la aplicación
7. Si está activo



The screenshot shows a 'User' pop-up window with the following fields and values:

Field	Value
Name	Agente Cinco
Email	agent5@bugtracking.es
Phone	620023383
Password	12345aA
Client	BugTracker
Role	Computer technician
Is active?	<input checked="" type="checkbox"/>

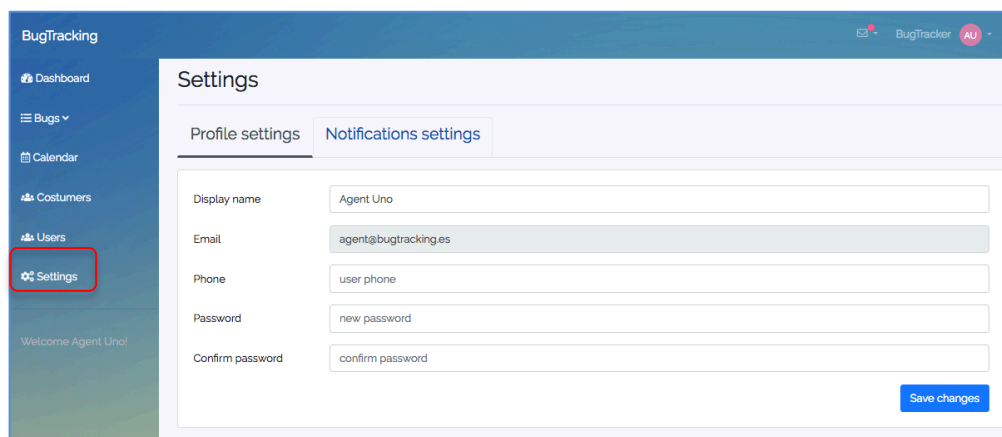
At the bottom right, there are two buttons: 'Close' and 'Save changes' (highlighted with a red rectangle).

Figura 63: Pop-up para registrar un nuevo usuario

6. Settings

En este apartado podemos ver como modificar la información de usuario y como ajustar las notificaciones.

Para acceder al apartado de Ajustes, haz clic en el menú lateral en la opción *Settings*.



The screenshot shows the 'Settings' page with a sidebar menu on the left. The 'Settings' option is highlighted with a red rectangle. The main content area has two tabs: 'Profile settings' (selected) and 'Notifications settings'. The 'Profile settings' tab contains the following fields:

Field	Value
Display name	Agent Uno
Email	agent@bugtracking.es
Phone	user phone
Password	new password
Confirm password	confirm password

A 'Save changes' button is located at the bottom right of the form.

Figura 64: Pantalla de ajustes del usuario

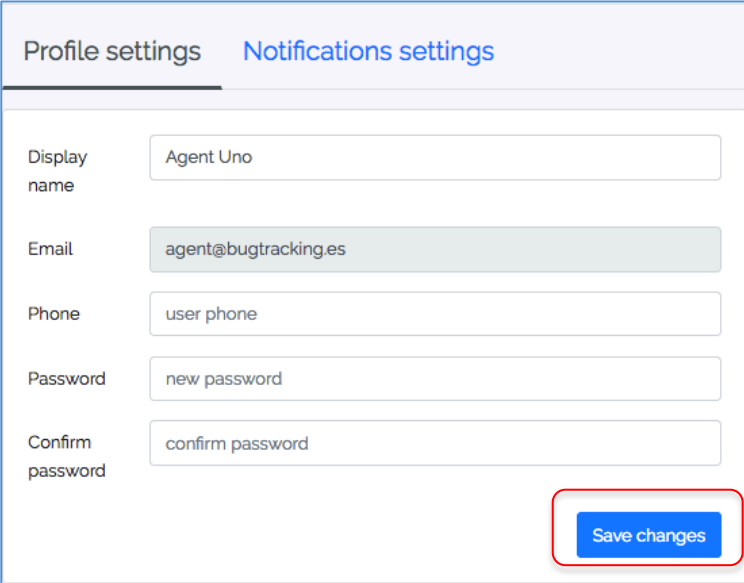
Modificar el perfil

Para ello, haz clic en la pestaña *Profile settings*, seguidamente se mostrará el formulario para modificar los datos de perfil:

1. Nombre que se mostrara en la aplicación
2. Teléfono
3. Contraseña

4. Confirmación de contraseña

Una vez introducido los campos, para guardar haz clic en el botón *Save changes*.



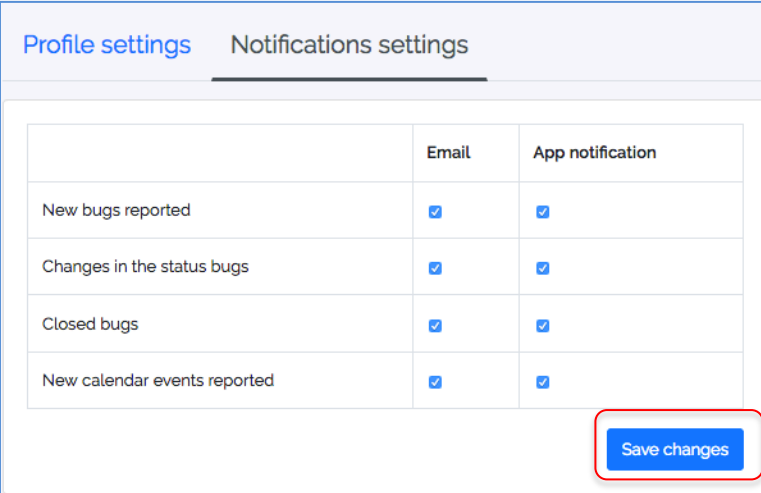
The image shows a 'Profile settings' form with two tabs: 'Profile settings' (active) and 'Notifications settings'. The form contains five input fields: 'Display name' (Agent Uno), 'Email' (agent@bugtracking.es), 'Phone' (user phone), 'Password' (new password), and 'Confirm password' (confirm password). A blue 'Save changes' button is located at the bottom right, highlighted with a red rectangle.

Figura 65: Formulario de edición del perfil

Modificar los ajustes de notificación

Para ello, haz clic en la pestaña *Notifications settings*, seguidamente se muestra en forma de tabla todos los ajustes de aviso que podemos activar / desactivar.

Para guardar los cambios, haz clic en el botón *Save Changes*.



The image shows a 'Notifications settings' form with two tabs: 'Profile settings' and 'Notifications settings' (active). The form displays a table with notification settings. The table has three columns: an unnamed column for the notification type, 'Email', and 'App notification'. All checkboxes are checked. A blue 'Save changes' button is located at the bottom right, highlighted with a red rectangle.

	Email	App notification
New bugs reported	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Changes in the status bugs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Closed bugs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
New calendar events reported	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 66: Formulario de ajuste de notificaciones

Manual para clientes

1. Acceder a la aplicación

Para hacer uso de la aplicación accede a la siguiente URL: <http://bt.accon.com/>

Seguidamente, introduce tu usuario y contraseña:

Usuario: lopez@medics.com

Contraseña: 12345aA

Una vez introducido los datos de acceso, haz clic en el botón *Login*.

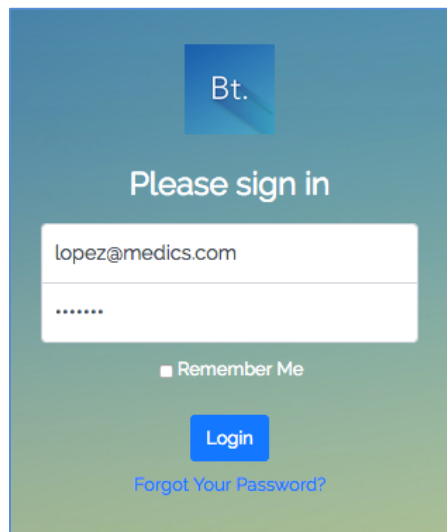
A login form with a blue header containing a 'Bt.' logo. The main heading is 'Please sign in'. Below it are two input fields: the first contains 'lopez@medics.com' and the second contains '*****'. A 'Remember Me' checkbox is located below the password field. At the bottom, there is a blue 'Login' button and a link that says 'Forgot Your Password?'.

Figura 21: Formulario de acceso a la plataforma

2. Crear una Incidencias

En este apartado se mostrará como registrar una nueva incidencia.

Para ello, en el menú lateral de la parte izquierda, haz clic en *Bugs*, y seguidamente en *Bug List*.

Una vez dentro, haz clic en el botón *New report*.

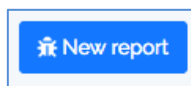


Figura 49: Botón para registrar una incidencia

A continuación, se mostrará el siguiente pop-up dónde introduciremos toda la información necesaria.

Los campos requeridos son:

1. Título: describir el título de la incidencia
2. Descripción: describir el problema de la incidencia.
3. SO: sistema operativo
4. Categoría
5. Prioridad
6. Reproducibilidad
7. Archivos adjuntos
8. Si es evento
 - a. Fecha inicio
 - b. Fecha fin

El resto de los campos no son obligatorios, pero se recomienda rellenarlos para así tener la máxima información de la incidencia.

Para finalizar, haz clic en *Save changes*.

Figura 50: Formulario para crear una nueva incidencia

3. Settings

En este apartado podemos ver como modificar la información de usuario y como ajustar las notificaciones.

Para acceder al apartado se Ajustes, haz clic en el menú lateral en la opción *Settings*.

Figura 64: Pantalla de ajustes del usuario

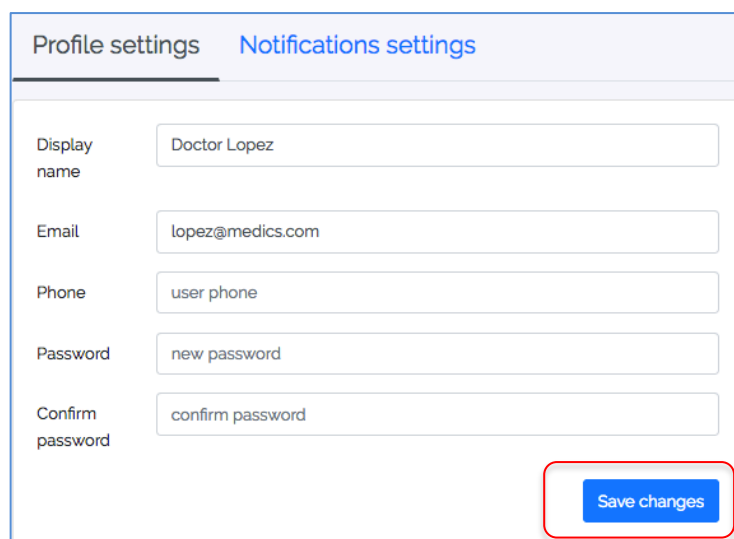
Modificar el perfil

Para ello, haz clic en la pestaña *Profile settings*, seguidamente se mostrará el formulario para modificar los datos de perfil:

1. Nombre que se mostrara en la aplicación

2. Teléfono
3. Contraseña
4. Confirmación de contraseña

Una vez introducido los campos, para guardar haz clic en el botón *Save changes*.



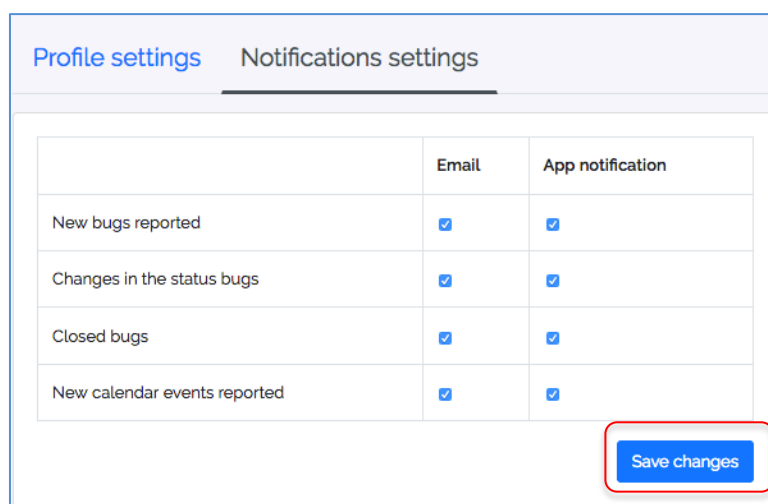
The image shows a 'Profile settings' form. It has two tabs: 'Profile settings' (active) and 'Notifications settings'. The form contains five input fields: 'Display name' (with 'Doctor Lopez'), 'Email' (with 'lopez@medics.com'), 'Phone' (with 'user phone'), 'Password' (with 'new password'), and 'Confirm password' (with 'confirm password'). A blue 'Save changes' button is at the bottom right, highlighted with a red border.

Figura 65: Formulario de edición del perfil

Modificar los ajustes de notificación

Para ello, haz clic en la pestaña *Notifications settings*, seguidamente se muestra en forma de tabla todos los ajustes de aviso que podemos activar / desactivar.

Para guardar los cambios, haz clic en el botón *Save Changes*.



The image shows a 'Notifications settings' form. It has two tabs: 'Profile settings' and 'Notifications settings' (active). Below the tabs is a table with 3 columns: an empty header column, 'Email', and 'App notification'. The table has 4 rows of notification settings. A blue 'Save changes' button is at the bottom right, highlighted with a red border.

	Email	App notification
New bugs reported	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Changes in the status bugs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Closed bugs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
New calendar events reported	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 66: Formulario de ajuste de notificaciones